Sommario

RUBRICHE

Direttore: Alessandro de Simone
Redazione/collaboratori: Claudio Baiocchi, Carlo e Lorenzo Barazzetta, Giovanni Bellù, Simone Bettola, Andrea e Alberto Boriani, Diego e Federico Canetta, Giancarlo Castanga, Pasquale D'Andreti, Maurizio Dell'Abate, Marco De Martino, Piero Dell'Orte, Luca Galluzzi, Michele Maggi, Giancarlo Mariani, Marco Miotti, Flavio Molinari, Claudio Mueller, Enrico Scelsa, D. Matturro, M.L. Nitti, Massimo Pollutri, Carla Rampi, Fabio Sorgato, Giovanni Verrelli, Antonio Visconti.
Segreteria di redazione: Maura Ceccaroli, Piera Perin
Ufficio Grafico: Mary Benvenuto, Arturo Ciaglia, Paolo Vertuccio
Direzione, redazione, pubblicità: V.le Famagosta, 75 - 20142 Milano - Tel. 02/8467348
Pubblicità: Milano: Leandro Nencioni (direttore vendite), Giorgio Ruffoni, Roberto Sghirinzetti (settore informatica), Claudio Tidone - V.le Famagosta, 75 - 20142 Milano - Tel. 02/8467348
● Emilia Romagna: Spazio E - P.zza Roosevel, 4 - 40123 Bologna - Tel. 051/236979
● Toscana, Marche, Umbria: Mercurio Srl - via Rodari, 9 - San Giovanni Valdarno (Ar) - Tel. 055/947444
● Lazio, Campania: Spazio Nuovo - via P. Foscari 70 - 00139 Roma - Tel. 06/8109679
Segretaria: Lilliana Degiorgi - **Abbonamenti:** Marina Vantini
Tariffe: prezzo per copia L. 3.500. Abbonamento annuo (11 fascicoli) L. 35.000. Estero: il doppio. Abbonamento cumulativo alle riviste Computer e Commodore Computer Club L. 70.000.
I versamenti vanno indirizzati a: Systems Editoriale Srl mediante assegno bancario o utilizzando il c/c postale n. 37952207
Composizioni: Systems Editoriale Srl - **Fotolito:** Systems Editoriale Srl
Stampa: La Litografica S.r.l. - Busto Arsizio (VA)
Registrazione: Tribunale di Milano n. 370 del 2/10/82 - Direttore Responsabile: Michele Di Pisa
Sped. in abb. post. gr. III - Pubblicità inferiore al 70% - **Distrib:** MePe, via G. Carcano 32 - Milano

**Che cosa riserva il futuro della tecnologia nel campo bellico, e quale è la produzione dei videogiochi e simulazioni di guerra col nostro Commodore?
Una serie di argomenti per divertirci oppure (a scelta) per passare notti insonni in preda al panico.**

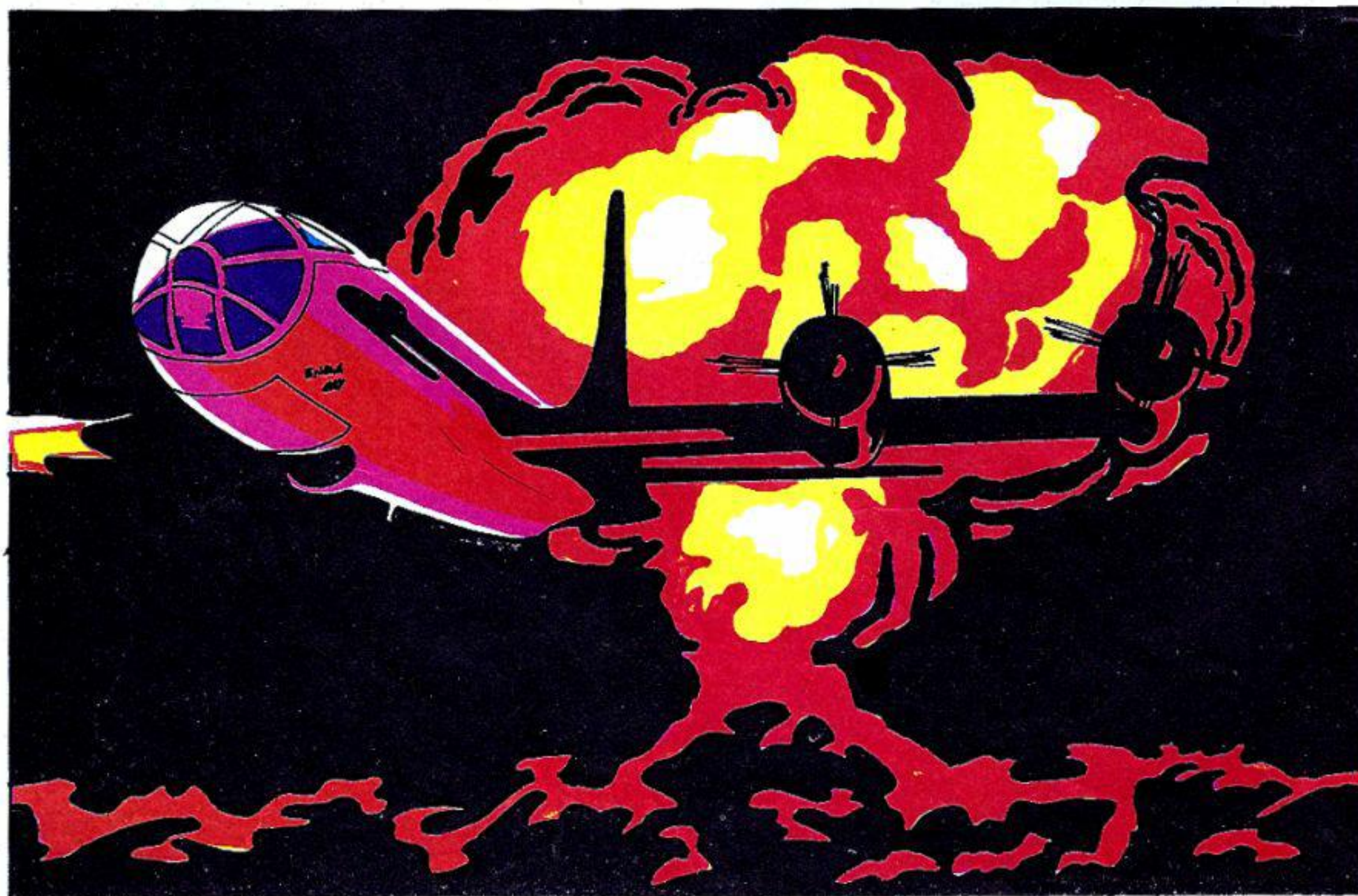
In altra parte della rivista troverete una serie di servizi su alcuni videogame che hanno riscosso un discreto successo tra gli appassionati di computer. Sono tutti, in un modo o in un altro, legati al mondo della guerra, alle simulazioni di conflitti da "dirigere" tramite la tastiera e, ad ogni modo, decisamente incruenti e, paradossalmente (dato l'argomento), divertenti.

Ma è di altro che ci preme parlare in queste colonne:

Ricordate la guerra delle Falkland (Malvine) in cui l'Inghilterra fu sfidata dall'Argentina per anacronistiche vicende di possessi territoriali?

Ebbene, dalle avvisaglie delle prime ostilità alla decisione dell'intervento britannico passarono diversi giorni, durante i quali le diplomazie di tutto il mondo cercarono, inutilmente, di metter pace tra i contendenti.

La fase successiva, cioè imbarco e viaggio via mare dei soldati di Sua Mae-

stà, richiese molti altri giorni per raggiungere i possedimenti sottratti alla Corona. Anche questi giorni furono spesi in febbrili trattative, puntualmente vane.

Poi, la guerra. Le solite vittime, da una parte e dall'altra, le solite medaglie alla memoria, alle quali avrebbero volentieri rinunciato gli sfortunati combattenti.

Il tempo, tuttavia, sarebbe risultato sufficiente se i diretti interessati avessero avuto realmente intenzione di evitare inutili perdite. La diplomazia, insomma, potè dire, in quell'occasione, la consolatoria frase di prammatica: "Abbiamo fatto tutto il possibile".

E tutto questo perchè, vivaddio, la distanza tra i due continenti era enorme, specie se da percorrere via mare.

Pensate invece ad altri tipi di interventi militari, a quelli, insomma, che giorno dopo giorno si perfezionano anche se non per via diretta, ma solo grazie a simulazioni di calcolatori.

Un missile intercontinentale impiega solo trenta minuti per andare, appunto, da un continente all'altro (o viceversa...). Se poi la base di partenza è rappresentata da un sottomarino atomico che incrocia il limite delle acque territoriali, il tempo si riduce a sette minuti. Solo 420 secondi separano la pressione del pulsante rosso dai megamorti nucleari.

Vi sono, insomma, reti di satelliti spia programmati per individuare, in qualunque parte del globo, punti termici "sospetti" che possono rappresentare le scie di eventuali, micidiali missili. Sembra che, bontà loro, siano in grado di distinguere con sufficiente approssimazione la differenza esistente tra i reattori di un normale aereo di linea e quella di un vettore intercontinentale. Inoltre sono "aiutati" dal fatto che una scia che sbuca all'improvviso dall'oceano non può che rappresentare un'unica alternativa.

Cosa c'entra tutto questo con i computer?

La risposta è banalmente agghiacciante: dati i tempi ristretti (sette oppure, al più, trenta minuti) non c'è tempo di radunare i ministri e i generali, per informare l'opinione pubblica, oppure per organizzare incontri diplomatici.

E' indispensabile (e sarebbe folle non farlo) computerizzare la rappresaglia atomica mediante un sistema automatico: individuazione missili eguale invio missili, magari in numero maggiore.

Sarebbe inutile, tra l'altro, ritardare la risposta perchè alcuni missili appartenenti all'attacco di sorpresa potrebbero esser fatti esplodere in volo per creare disturbi elettro—magnetici che impedirebbero il puntamento e la partenza dei missili di rappresaglia. Il tempo di risposta, quindi, deve esser contenuto nella metà, prudenzialmente, di quelli visti prima.

Si dice che, in ogni caso, la pressione del pulsante "definitivo" spetta all'uomo.

E se qualcuno di buon senso, lanciato l'allarme, si rifiutasse di premerlo? E se, magari, alcuni impianti di allarme fossero posti fuori uso dalle prime bombe o da squadre di sabotatori, come intervenire sui delicati circuiti di lancio?

La risposta, volendo considerare tutte le eventualità possibili, deve esser trovata inevitabilmente attraverso un sistema di lancio in grado di esser attivato per vie molto semplici.

E qui si entra nel segreto militare, da una parte, e nella fantaguerra, dall'altro.

Attraverso un comune telefono è possibile, grazie a un codice, inviare...?

E allora perchè alcuni ragazzi americani, col proprio personal computer e un semplice modem, sono riusciti ad entrare in zone Top Secret rigorosamente protette?

Dobbiamo forse ritenere che nessuno ha finora pensato a realizzare linee telefoniche militari separate completamente da quelle civili?

Oppure ci è lecito immaginare che, in caso di estrema necessità, è doveroso prevedere il "rifiuto" di qualcuno addetto ai lanci e, quindi, di "bypassarlo" intervenendo anche in questi casi?

Magari accendete una sigaretta, e riflettetici su. Perchè proprio una sigaretta?

Forse perchè, in media, dura sette minuti...

Alessandro de Simone

AVVERTENZA IMPORTANTE

Tra le numerose domande che pervengono in Redazione, vengono scelte in prevalenza quelle di interesse generale, rinviando, spesso sinedie, la pubblicazione di quelle ritenute meno interessanti oppure già affrontate di recente su Commodore Computer Club.

Ci capita infatti, purtroppo, di esaminare schede in cui vengono richiesti argomenti già trattati sullo stesso fascicolo da cui la scheda è tratta!

Invitiamo pertanto i lettori, specie se principianti, a sfogliare con attenzione le pagine della nostra rivista e a leggere articoli che, pur se apparentemente non li interessano, contengono purtuttavia preziose informazioni su istruzioni particolari (POKE, PEEK, SYS ed altre) o tecniche insolite di programmazione, registrazione di dati, programmi eccetera.

Da nastro a disco

☐ Possiedo molti programmi acquistati su nastro e, ora che ho comprato il drive 1541, mi piacerebbe "passarli su disco". Come fare per effettuare il travaso senza incorrere nel blocco delle numerose protezioni?
(Numerosi lettori)

• Apprezziamo sempre il modo garbato con cui molti lettori chiedono la pubblicazione di sistemi per sproteggere.

A parte il fatto che abbiamo più volte affermato di non aver alcuna intenzione di favorire il mercato della pirateria software (non ce ne vogliano coloro che sono in buona fede), la questione, presentata sotto molteplici forme, è di difficile soluzione.

I sistemi di protezione (e conseguente sprotezione) sono pressochè infiniti e, di conseguenza, indicarne solo qualcuno non risolverebbe il problema.

Per quanto riguarda il software da noi prodotto (Systems Editoriale) tra breve dovremmo commercializzare su disco (ma solo per posta) i programmi della nostra cassetta "Software Club" e (perchè no?) quelli che vedete pubblicati su C.C.C.

Cartucce e fastidi

☐ Inserendo, sul retro del computer, cartucce di utility, sorgono inconvenienti relativi al caricamento di alcuni programmi, oppure al loro corretto funzionamento. Come mai?
(Orlando Proietti — Roma)

• Molte di queste cartucce, non appena il computer viene acceso, "riempono" alcune locazioni RAM con brevi segmenti di programma in Linguaggio Macchina (L.M.) Naturalmente l'utilizzatore non si accorge minimamente di tale operazione e lavora col calcolatore come di consueto. Particolari programmi, però, occupano proprio quelle locazioni di memoria ed è evidente che sorge una conflittualità che porta, molto spesso, all'"inchiodamento" del sistema.

Con queste cartucce, a volte, la stessa routine che sovrintende al caricamento dei programmi viene modificata con la conseguenza che molti programmi protetti generano confusione nella fase di caricamento.

L'unico modo per venirne a capo è quello di individuare i programmi che presentano gli inconvenienti lamentati, e di caricarli solo dopo aver rimosso la cartuccia.

Ancora sulle Banche Dati

☐ In seguito a numerose domande dei lettori, precisiamo (ripetendoci...) quanto segue:

• E' possibile collegare via telefono il proprio computer con altri computer (o banche dati) per mezzo dell'accessorio chiamato Modem il cui costo oscilla tra le 200mila e il milione di lire.

• In genere il software necessario per utilizzare il modem è fornito insieme con l'apparecchio.

• Per il collegamento è sufficiente disporre di un computer con porta seriale RS—232 (o compatibile).

• E' indispensabile, per collegarsi con una banca dati, entrare in precedenza in contatto con questa e, pagando un canone di abbonamento, ricevere il codice segreto di identificazione per mezzo del quale entrare in contatto.

• Il costo del telefono è solo quello relativo al numero di scatti effettuati durante il collegamento. Se la linea è disturbata, tanto da costringere l'utente ad un nuovo tentativo, gli scatti si sommano ai precedenti. Morale: prepararsi a bollette salate.

• Non ci risulta che ci siano banche dati di un certo interesse per i nostri lettori. Ecco perchè non ne parliamo mai.

Una strana freccia

☐ In alcuni vostri listati compaiono comandi ed istruzioni Basic, preceduti da una freccia, che non ho visto in nessun altro programma. Che cosa rappresentano?
(A. Sasso — Bari)

• I computer Commodore sono dotati del linguaggio Basic denominato V2. Questa sigla lo contraddistingue dalle tante versioni Basic che, finora, sono state inserite nei vari computer.

Grazie, però, alla manipolazione di particolari zone di memoria, è possibile "creare" nuove istruzioni in grado di svolgere compiti particolari, soprattutto grafici o sonori.

E' ovvio che bisogna fare in modo che il computer riconosca i nuovi comandi (pena l'emissione di un Syntax Error) e si comporti di conseguenza.

Una delle tecniche usate dai programmatori, è proprio quella di aggiungere un carattere, usato di norma con scarsa frequenza, al nome della nuova funzione. Il calcolatore, quando lo incontra durante un'elaborazione, salterà ad eseguire le routine in Linguaggio Macchina che sovrintendono quella particolare funzione. Uno dei caratteri—codice più frequentemente usato è, appunto, il carattere di freccia a sinistra, utilizzato per le routine grafiche da D.Toma, uno dei nostri più validi collaboratori.

Per utilizzare i comandi Basic "aggiunti" è necessario, quindi, dapprima caricarli nella memoria del computer e, in seguito, digitare o caricare i programmi che le utilizzano.

Tutto questo, però, era spiegato nell'articolo che citi e ci meraviglia che ti sia sfuggita una nota così importante.

Uso dei tasti funzione

☐ Come è possibile utilizzare nei propri programmi i tasti funzione F1...F8?
(D. Crivellotto — Mediglia)
(G. Esposito — Firenze)

• I quattro tasti funzione ubicati alla destra dei calcolatori Commodore hanno una proprietà molto importante in un computer moderno: l'ergonomia.

Volendo effettuare, infatti, una delle possibili scelte di un menu, l'utilizzatore, specie se principiante, è facilitato nell'individuare un tasto posizionato in una zona della tastiera rapidamente raggiungibile.

Si potrebbe raggiungere lo stesso identico risultato se, invece di un messaggio del tipo "Premi il tasto A per continuare" si visualizzasse "Premi il tasto F1 per...".

La differenza, ripeto, è solo di natura ergonomica, vale a dire di "comodità" per l'utilizzatore del computer.

E' ovvio che bisogna creare un ciclo di attesa (ricorrendo all'istruzione GET) per fare in modo di individuare il tasto premuto. Un esempio valido per capire quale dei tasti è stato digitato, può essere il seguente:

```
100 GETA$:IFA$=""THEN100
110 IFA$="A"THEN 200
120 IFA$="B"THEN 300
```

eccetera.

Si può anche ricorrere, per motivi di chiarezza, al codice ASCII del tasto premuto. Questo, lo ricordiamo, associa molti dei valori compresi tra 0 a 255 a ciascun carattere della tastiera compresi quelli "speciali" dei colori, posizionamento cursore, tasti funzione, caratteri semigrafici eccetera.

Fate girare il seguente mini programma per individuare il codice di ciascun tasto e, durante il suo funzionamento, premete i vari tasti della tastiera: vedrete visualizzato, dopo ogni pressione, il codice ASCII corrispondente:

```
100 GETA$:IFA$=""THEN100
110 PRINTASC(A$):GOTO100
```

I risultati di questo esame potete utlizzarli, ad esempio, in un vostro menu:

```
100 GETA$:IFA$=""THEN100
110 IFASC(A$)=65THEN200
120 IFASC(A$)=66THEN300
```

eccetera.

Un monitor dispettoso

□ **Da quando ho comprato il monitor 1702 non riesco a caricare, da nastro, nessun programma se non spegnendo il monitor e riaccendendolo a caricamento effettuato. Come mai questo inconveniente non mi capitava usando un normale apparecchio TV?**
(Roberto Fabbretti — Ciriè)

• Quando un TV (o apparecchio similare come un monitor) è in funzione, genera campi elettrici e magnetici che influenzano la zona circostante. Durante il "colloquio" tra computer e registratore, i dati e i programmi sono caricati grazie a fenomeni di natura elettrica e magnetica.

E' molto probabile, quindi, che il campo elettro—magnetico del monitor influenzi negativamente quello del tuo registratore. Per porre rimedio all'inconveniente puoi provare uno dei seguenti sistemi, suggeriti, del resto, molto spesso attraverso la nostra rivista:

• Allontanare quanto più possibile il monitor dal registratore e dal suo cavo, almeno durante le fasi di lettura o registrazione.

• Inserire uno schermo metallico tra monitor e registratore. Un foglio di alluminio, di quelli adoperati normalmente per la conservazione dei cibi, è adattissimo allo scopo.

• Collegare ad un impianto di messa a terra (come un termosifone) il cavetto metallico di massa che fuoriesce dalla presa del registratore.

Monitor diversi

□ **Come mai il monitor 1901 acquistato nel dicembre '85 è diverso da quello che compare nei depliant pubblicitari della Commodore?**
(S. Calzolai — Prato)

• Purtroppo non ci invii una foto dell'esemplare in tuo possesso e ci è difficile darti una risposta precisa.

E' molto probabile, comunque, che la differenza riguardi solo l'aspetto estetico e non la circuiteria elettronica dell'apparecchio. Capita spesso, infatti, che una fabbrica decida di cambiare il design dei suoi prodotti allo scopo di valutare il favore del pubblico.

Se, comunque, c'è il marchio Commodore non hai nulla da temere sia per ciò che riguarda la garanzia sia per ciò che riguarda la qualità del monitor.

Numeri senza spazi

□ **Al comando PRINTVAL("48") il computer stampa il numero 48 spostato di uno spazio a destra. Come potrei assegnare ad una stringa un valore numerico senza lo spazio che lo precede sempre?**
(G. Sanson — Trieste)

• Lo spazio che precede sempre un valore numerico è dovuto al segno, che può esser positivo oppure negativo.

Se è positivo, il segno viene sottinteso ed il computer si limita a stampare, al suo posto, uno spazio bianco. Nel caso il valore sia negativo, invece, non è possibile sottintenderlo ed il computer è "costretto" a visualizzarlo. Prova, infatti, a digitare:

```
PRINT48:PRINT—48
```

Se, ora, assegni ad una stringa qualunque (come A$) il compito di trasformare il numero in un gruppo di caratteri numerici, anche il segno viene trasformato in carattere e memorizzato, o meno, come spazio bianco. Prova a digitare le seguenti righe:

```
A=48:A$=STR$(A):PRINTA$
A=—48:A$=STR$(A):PRINTA$
```

In entrambi i casi vedrai comparire il numero (o meglio: i caratteri) 48 preceduti da uno spazio bianco (segno positivo) o dal trattino (segno negativo).
Per ottenere soltanto i caratteri numerici, privi cioè del segno, è sufficiente aggiungere la funzione stringa RIGHT$ che "restituisce" gli N caratteri di destra di una stringa qualunque.
Prova a digitare:

```
100 A=48:A$=STR$(A)
110 B$=RIGHT$(A$,LEN(A$)—1)
120 PRINT B$
```

In questo modo, qualunque sia il valore numerico assegnato ad A, la stringa A$ conterrà un carattere (il primo a sinistra) rappresentante il segno. Se, ricorrendo alla funzione RIGHT$, "escludiamo" tale carattere, rimangono solo i caratteri numerici. Attento, però: in questo modo non tieni conto del segno negativo eventualmente presente.

per i **PROGETTI SPECIALI** della Systems Editoriale

collaboratori a tempo pieno oppure parziale.

La persona ideale:

- risiede a Milano o nel suo hinterland;
- è **REALMENTE** esperta in Linguaggio Macchina e Assembler;
- è in grado di sviluppare autonomamente programmi di qualsiasi tipo;
- è in possesso di un sistema completo (computer Commodore, drive, registratore, ecc.);
- è interessato ad acquisire esperienza anche su altri sistemi (MSX, Sinclair, MS/DOS, ecc.)

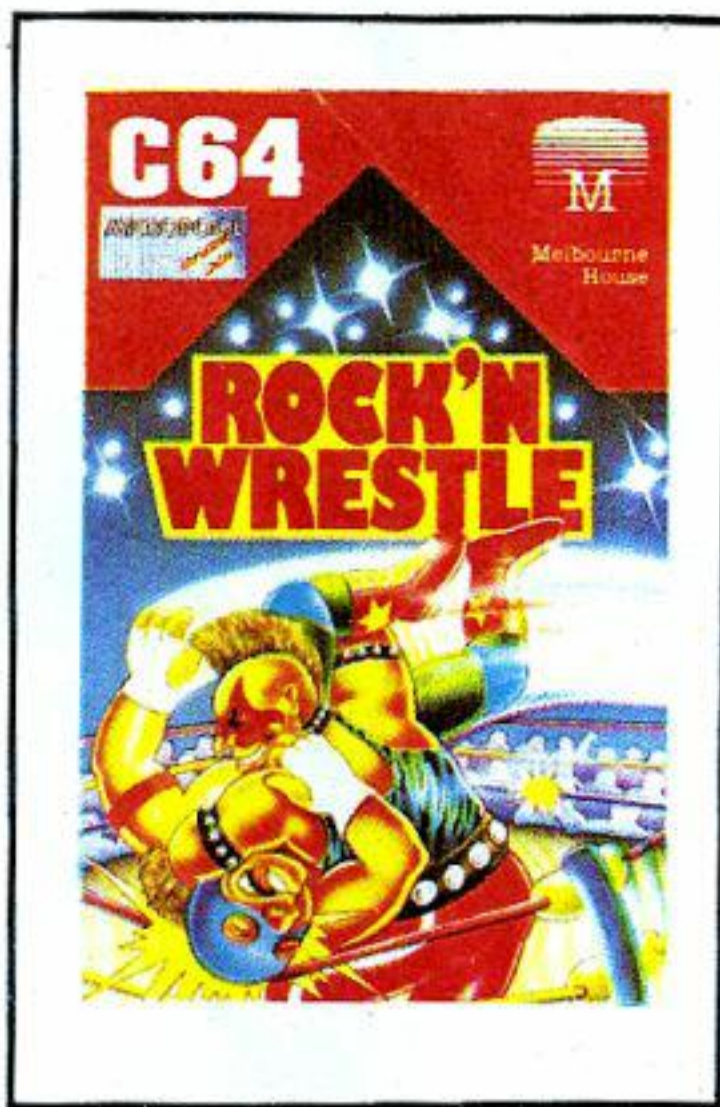
Per un primo contatto **TELEFONARE** alla Systems Editoriale (tel. 8467348) il martedì o venerdì pomeriggio (dopo le 16) chiedendo dell'ingegner de Simone.

Rock'n Wrestle

Se siete abili nel manovrare il joystick, questo gioco è per voi

Un nuovo game della serie "sportiva" che ricorre intensivamente agli sprite per creare animazioni davvero realistiche.

Lo sport, che i nostri lettori sicuramente hanno visto qualche volta nel corso di tramissioni televisive, consiste nella lotta spettacolare tra due energumeni. Lo scopo è quello di mandare al tappeto l'avversario ricorrendo a "prese" di ogni tipo. Non è inconsueta, tra le altre mosse, quella che consente di afferrare il lottatore, farlo girare un paio di volte per poi scaricarlo brutalmente sulla pedana.

Si può giocare da soli contro il computer, oppure in due giocatori e, a seconda dei casi, è necessario disporre di uno solo oppure di due joystick. Chi lo desidera può comunque limitarsi alla tastiera, il cui uso è previsto, ma è risaputo che nei giochi d'azione nulla può sostituire i joystick. Il movimento di questi ultimi, come è intuitivo, risulta complesso dato che, oltre agli spostamenti nelle varie direzioni, molte sono le "mosse" che è possibile impostare. Il videogame di cui si parla è pertanto adatto a chi non si accontenta più dei soliti giochi dal controllo facile, ma richiede abilità particolare nel premere contemporaneamente, o meno, il pulsante del fire per selezionare una delle "prese" o per sferrare un attacco contro l'avversario.

I lottatori che entrano in azione sono numerosi e ognuno sembra dotato di abilità particolari. Le schermate piene di colori, la musica di sottofondo, alcune parole sintetizzate dell'arbitro, l'effetto "folla", rendono questo gioco decisamente piacevole e impegnativo.

Non è difficile portare a termine una partita ed accorgersi che la mano che manovra il joystick si è indolenzita, come se avessimo realmente partecipato alla lotta...

Express System

Gestione dei dischi ad alta velocità e basso... prezzo

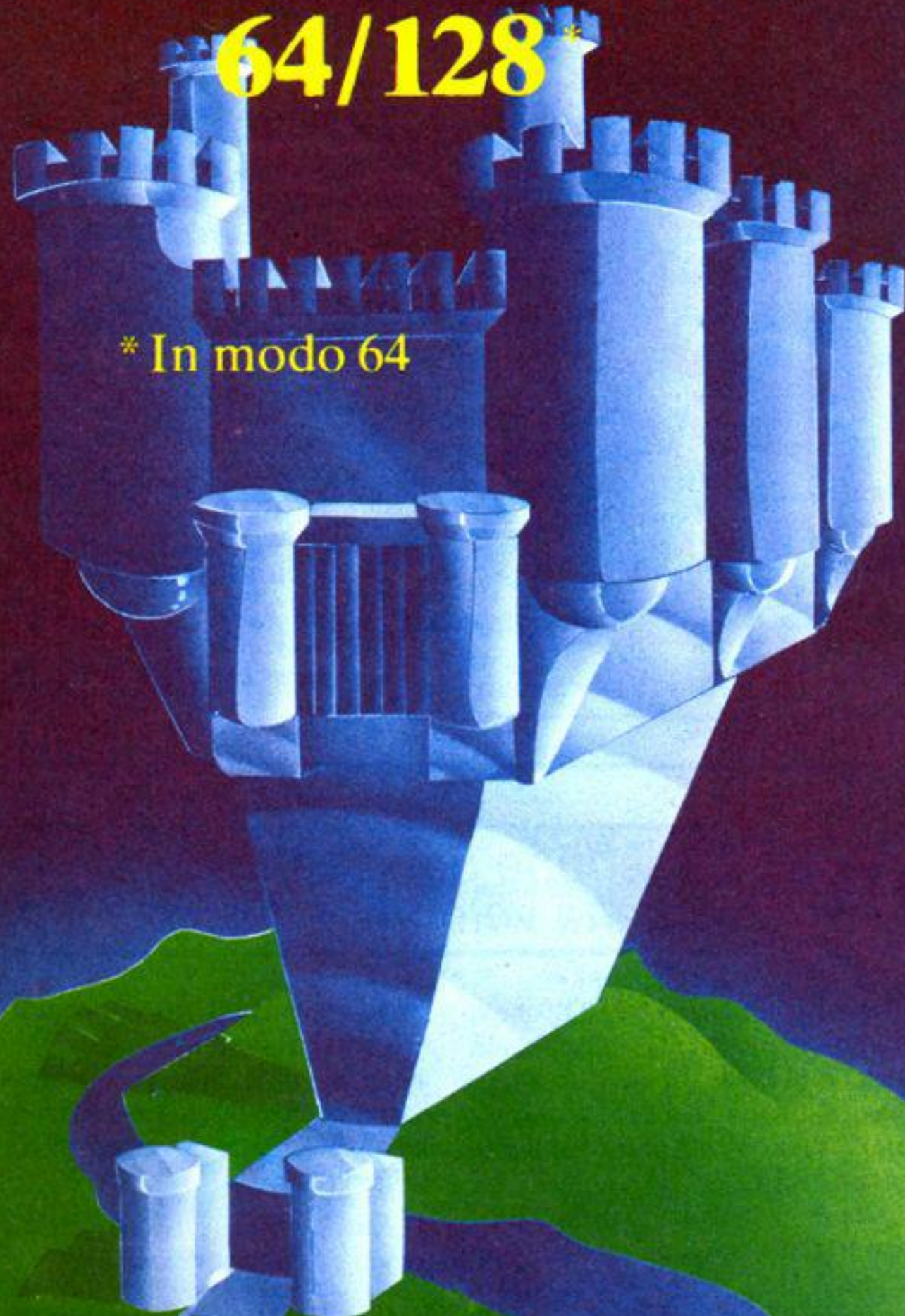
La nuova cartuccia per C64, commercializzata dalla Mastertronic, consente di effettuare le operazioni di colloquio col disco con una velocità che può risultare anche cinque volte maggiore del consueto.

Un programma di 70 blocchi, ad esempio, che richiede un tempo di caricamento di ben 45 secondi, con la cartuccia inserita viene caricato in meno di 10 secondi.

Non appena si accende il computer, i tasti funzione risultano programmati ed è possibile utilizzarli immediatamente:
F1: esame della Directory.
F2: compare il comando LOAD" che si riferisce automaticamente al disco e non richiede quindi il ",8" finale.
F3: sposta il cursore in alto
F4: compare, anch'esso riferito al disco, il comando "SAVE"
F5: carica e fa partire il primo programma presente sul disco.
F6: esegue il LIST del programma presente in memoria.
F7: disabilita la cartuccia, ad eccezione delle routine di caricamento veloce, e manda in esecuzione il programma.

Questo accorgimento consente di caricare e di far partire gran parte dei programmi che richiedono alcune locazioni occupate dalla cartuccia evitando conflittualità di funzionamento.
F8: viene disabilitata completamente la cartuccia evitando di rimuoverla dal suo alloggiamento nel caso non si desideri utilizzarla.

Altri comandi sono disponibili ricorrendo ad opportuna sintassi.

Ciò che rende particolarmente interessante la cartridge è il fatto che lo schermo non viene "cancellato" durante il caricamento, come invece normalmente capita usando programmi velocizzatori. Il prezzo di listino è piuttosto contenuto (L. 49.000 Iva compresa).

Una cassetta tutta nuova

di Valerio Ferri

La Systems Editoriale presenta il nuovo numero della cassetta "Software Club", che, come sempre, include numerosi giochi nuovi e divertenti per utenti di Commodore 64, Commodore 16, Vic 20, ZX Spectrum e Msx.
Vediamoli insieme dettagliatamente:

Commodore 64

JAN SOLO: avventura spaziale ispirata al famoso film "Guerre Stellari". In questo game del genere "Adventure" voi interpretate il ruolo dell'avventuriero dello spazio Jan Solo, che nel film era, come ricorderete, interpretato da Harrison Ford.

VIETNAM: come quasi tutti i videogames, anche questo è ispirato ad un celebre film: "Apocalypse now". A bordo di un elicottero dovrete eliminare tutti i velivoli nemici e... attenzione al carburante.

STAR TREK: una miscela di strategia e azione fa di questo prodotto un gioco molto avvicente.
L'Enterprise, astronave interstellare della Confederazione dei Pianeti Uniti, ha il compito di ripulire lo spazio intergalattico dai perfidi Klingons.
Dovrai impersonare il capitano Kirk nella distruzione degli alieni.

BIO INVADERS: "Viaggio allucinante" è invece il film al quale si ispira quest'ultimo gioco.
Per portare a termine la missione avete in dotazione tre micro sommergibili, da controllare per mezzo della tastiera, con i quali dovrete rompere la membrana citoplasmica della cellula.

SWAP: Un'utility di notevole interesse che consente di mantenere in memoria fino a tre programmi Basic da 12 Kbyte l'uno con possibilità di selezionarne uno a scelta mediante i tasti funzione.
Ideale per apprendere nuove tecniche di programmazione e di protezione.

Commodore 16

DAMA: finalmente il nostro calcolatore

ha imparato a giocare a dama.
Le regole rispecchiano fedelmente quelle del classico gioco la cui strategia, come è noto, non può essere sottovalutata.

IDENTIKIT: siete l'unico testimone oculare di un omicidio e vi è stato affidato il compito di riconoscere l'assassino utilizzando il solo metodo dell'identikit. Programma interessante per il ricorso intensivo ai caratteri ridefiniti.

BIO INVADERS: versione funzionante su Commodore 16 analoga a quella già menzionata per il C64.

LOTTO: il gioco del sabato è simulato perfettamente con il vostro computer per cui... occhio al 34 sulla ruota di Napoli!

Vic 20

DANGER: l'espansione da 16 KRam, oltre che al joystick, sono gli accessori necessari per questo gioco. A bordo del vostro aereo dovete distruggere i pericolosi invasori spaziali che cercano di attaccarvi. Notevole l'effetto grafico realizzato grazie a velocissimi scroll laterali.

GRAPHIC ROUTINES. Udite! udite! Le famose routine grafiche di Danilo Toma, realizzate solo per il C64, sono ora disponibili sul VIC 20 sebbene richieda l'espansione da 16 KRam. D'ora in poi non avrete più limitazioni per disegnare in alta risoluzione grazie alla strabiliante velocità di esecuzione.

MSX

CHECKERS: è in tutto e per tutto il famoso gioco della dama, elettronicamente tradotto. Come temibile avversario avete, ovviamente, il vostro msx.

VORTEX: basato su un classico passatempo da tavolo, questo gioco mette a dura prova la vostra intelligenza.

Zx Spectrum

AMBULANCE: è una situazione veramente difficile: dovete impedire che le macchie d'olio sparse sulla strada da impuniti mascalzoni, facciano sbandare le macchine che sopraggiungono provocando danni ed incidenti.

PUZZLE: eccoci ora a risolvere un semplice puzzle numerato... Si tratta di unire con la vostra penna computerizzata i vari punti nella sequenza indicata. La penna, come di consueto, viene manovrata tramite tastiera.

FOOTING: questa mattina vi siete alzati con il sano desiderio di effettuare un pò di footing, al fine di mantenervi in splendida forma.
Effettuerete dapprima una corsa in campagna, poi passerete a correre sull'arenile di una splendida spiaggia. La corsa termina in città.
Purtroppo non dovrete semplicemente fare i conti con la vostra resistenza fisica, ma anche con numerosi ostacoli.

INVASIONE TOTALE: per conquistare le tre fortificate città di Parigi, New York e Roma, dovrete combattere contro un cannone laser, sempre più efficiente a mano a mano che passa il tempo.
Se però riuscirete a superarlo nelle tre prove, il mondo sarà vostro!!!

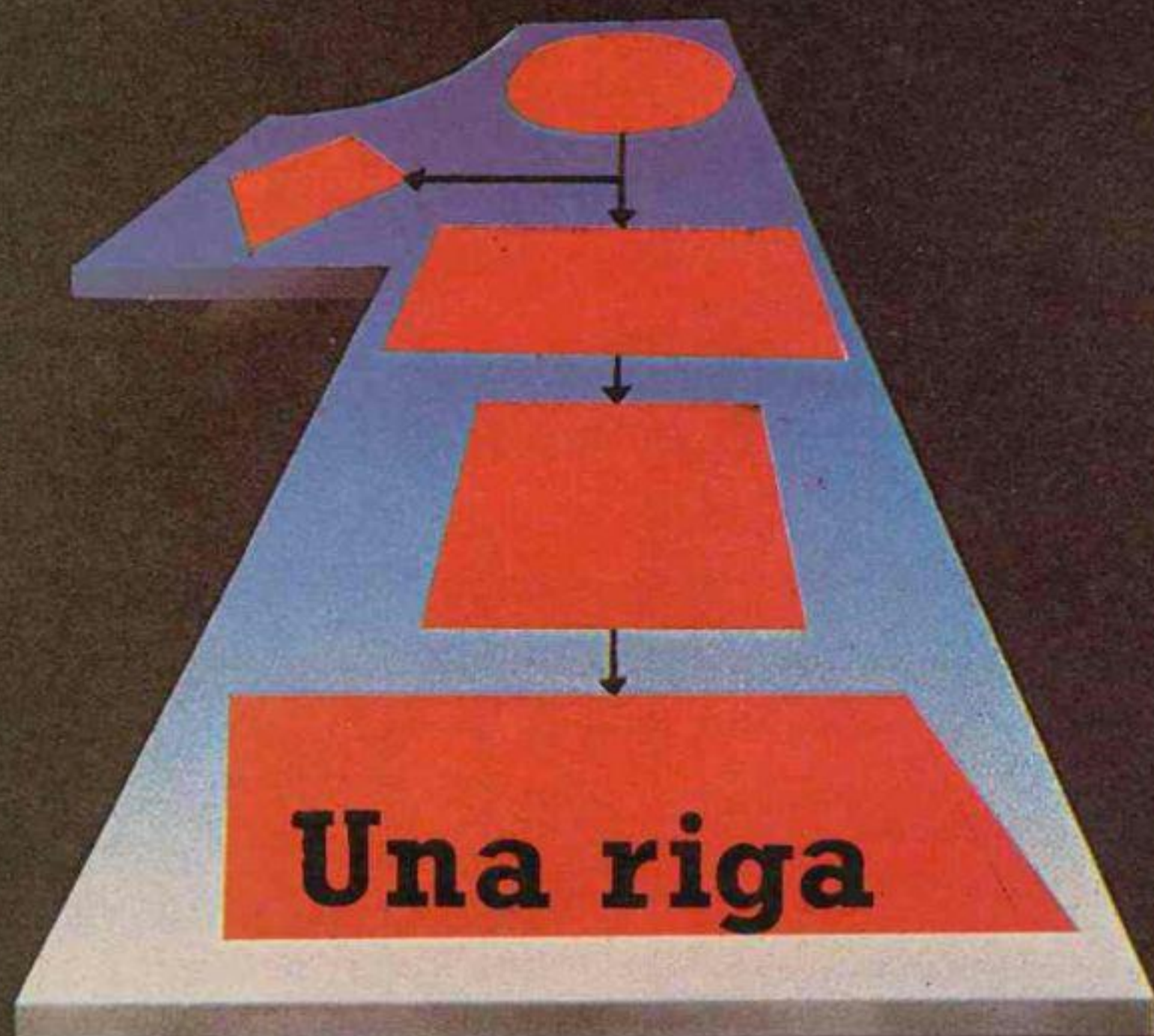
Novità

Il programma IDENTIKIT inserito nella cassetta, è protetto in un modo piuttosto semplice.
Ai primi 10 lettori che comunicheranno il modo in cui abbiamo applicato la protezione (e NON il sistema per sproteggerlo) verranno selezionate le risposte più precise e dettagliate. L'ovvia conseguenza, come potete intuire, sarà costituita dall'invio di materiale della Systems editoriale (cassete, libri, eccetera).

Computer table

Un hobbysta, molto spesso, ha problemi di spazio, soprattutto quando decide di acquistare altre periferiche da collegare al proprio personal.

La ditta Evergreen di Cornaredo (Mi) (Tel. 02/9362538) propone un mobile contenitore che, al vantaggio di occupare poco spazio, aggiunge quello di esser trasportabile su rotelle. L'antina verticale di cui dispone si trasforma, una volta aperta, in comodo ripiano su cui appoggiare il computer. È possibile aggiungere un braccio snodabile su cui sistemare un monitor mentre i cavi di collegamento possono fuoriuscire dallo schienale opportunamente forato.

Quando non si lavora col computer si ripongono le apparecchiature all'interno del contenitore che, trasportato ovunque, trova di sicuro facile sistemazione in un angolo della camera.

Su ogni numero di Commodore Computer Club compaiono un paio di pagine dedicate a una dozzina di micro programmi lunghi una sola riga.

E' ovvio che non è possibile pretendere effetti sorprendenti in listati così brevi, tuttavia, sviluppando l'idea su cui gli stessi programmi son basati è possibile pervenire a realizzazioni di tutto rispetto ricorrendo, è inutile dirlo, a qualche riga in più.

Particolarmente interessanti per i principianti, spesso utili anche per gli esperti, i micro programmi di una sola riga rappresentano una valida "palestra" per abituarsi a concentrare in poche, essenziali istruzioni la soluzione di problemi complessi solo in apparenza.

◇◇◇◇◇◇◇◇◇◇◇◇

Ecco una valanga di "una riga" (ben trenta mini listati) tutti nuovi e specifici per il Commodore 64. Molti di questi, tuttavia, possono "girare" con minime modifiche sugli altri Commodore.

Ad inviarli è un simpatico lettore di Genova (Roberto Sagoleo) sicuramente in gamba e ricco di fantasia. L'originalità dei programmi la dedichiamo ironicamente a chi, in pieno '86, osa ancora accusare gli hobbisti del computer di mancanza di elasticità mentale.

1

Pianoforte. La tastiera del computer si trasforma in un pianoforte dato che, alla pressione di ciascun tasto, corrisponde l'emissione di una nota musicale. Premendo il tasto Run/Stop il programma si interrompe e una nota viene emessa di continuo. Se, invece, tenete abbassato il tasto Shift/Lock, ad ogni pressione di Run/Stop verrà emessa una nuova nota senza che il programma si interrompa. Originale è l'idea di ricorrere al ciclo For...Next con Step nullo, che consente di realizzare un ciclo senza fine.

```
1 L=54273:POKE L+5,
  128:POKE L+23,15:
  POKE L+3,33:FOR I
  =Ø TO 1 STEP Ø:PO
  KE L,PEEK(197) AN
  D 63:NEXT
```

2

Curve sinusoidali. Un asterisco disegna una (quasi) infinità di curve sinusoidali l'una sotto l'altra.

```
1 FOR X=Ø TO 39:PRI
  NTCHR$(19):FOR R=
  Ø TO (9+9*SIN(X/9
  *π)):PRINT:NEXT:P
  RINT TAB(X+Y)"*":
  NEXT:Y=Y+1:GOTO 1
```

3

TV Color. Questo programma consente la regolazione cromatica del vostro TV Color oppure, se possedete un TV in bianco e nero, di individuare più facilmente i colori che riproducono la stessa tonalità di grigio.

```
1 FOR I=Ø TO 39:PRI
  NTCHR$(18)CHR$(32
  );:NEXT:POKE 646,
  15 AND PEEK(646)+
  1:GOTO 1
```

4

TV Audio. Prima di caricare programmi sonori, fate girare questa riga in modo da regolare opportunamente il volume del vostro TV. Il mini programma, inoltre, si presta per essere utilizzato come effetto sonoro nei vostri programmi Basic.

```
1 L=54273:POKE L+23
  ,15:POKE L+5,128:
  POKE L+3,33:FOR V
  =+Ø TO 1 STEP Ø:F
  OR I=Ø TO 255 STE
  P .5:POKE L,I:NEX
  TI,V
```

5

Ufo. Siete alla guida di un Ufo che lascia dietro di sè una scia. Inserendo il joystick nella porta N. 2, si fa partire il programma con PRINT CHR $ (147): RUN. Chi possiede un vecchio modello di Commodore 64 deve digitare, dopo aver battuto e verificato la riga proposta, le seguenti istruzioni:

PRINT CHR $ (147): POKE 53281,1:RUN

Notare la formula matematica che consente di spostare la pallina in accordo con i movimenti del joystick. La notevole brevità del listato, e la conseguente rapidità di risposta, invoglia a creare videogiochi molto interessanti.

Purtroppo, in una sola riga, non è stato possibile inserire anche un controllo sui limiti dello schermo. Ciò vuol dire che se si supera il bordo inferiore del video, l'Ufo piomba nell'area destinata al Basic e... ne combina di tutti i colori!

```
1 Q=PEEK(56320):Y=Y
  +(Q AND 1)-(Q AND
   2)/2:X=X+(Q AND
  4)/4-(Q AND 8)/8:
  POKE 1024+X+Y*40,
  81:GOTO 1
```

6

Bang! . Usate il grilletto del vostro joystick (da collegare in porta 2) come una vera pistola con un effetto sonoro davvero sorprendente. Se, durante il funzionamento udite un sibilo acuto oppure altri effetti sonori oltre quello dello sparo, dovete azzerare tutti i registri prima di utilizzarlo, oppure spegnere e riaccendere il calcolatore.

```
1 L=54296:POKE L,15
  :POKE L-5,9:POKE
  L-9,99:FOR I=0 TO
   1 STEP 0:POKE L-
  6,130+(PEEK(56320
  )=111):NEXT
```

7

RUN in italiano. Un messaggio made in Italy non fa mai male: per far girare un qualsiasi programma, digitato DOPO aver lanciato questa riga, è necessario digitare VAI e non più RUN. Attendere qualche secondo per la trascrizione delle diverse migliaia di dati (da 40960 a 49151).

```
1 FOR I=40960 TO 49
  151:POKE I,PEEK(I
  ):NEXT:POKE 1,54:
  I=41158:POKE I-1,
  86:POKE I,65:POKE
   I+1,201
```

8

Colori. Digitate questo programma per esaminare con calma le possibilità cromatiche dell'incredibile Commodore 64.

```
1 POKE 53265,0:FOR
  I=0 TO 16 STEP .0
  1:POKE 53280,I:NE
  XT:POKE 53265,27
```

9

Prova joystick. Facendo girare questo programma con il joystick inserito in Port 2 compariranno di continuo cinque valori affiancati. Il numero zero indicherà quale dei cinque interruttori del joy (fuoco, alto, basso, destra, sinistra) è attivato in quel momento. Utile per individuare il tasto premuto nei vostri videogame.

```
1 Q=PEEK(56320):FOR
   I=4 TO 0 STEP -1
  :PRINT(Q AND 2↑I)
  >0;:NEXT:PRINT:GO
  TO 1
```

10

Trenino. Un simpatico trenino attraversa sbuffando lo schermo. Per capire meglio i caratteri da digitare, tenete presente che, dopo l'apertura delle virgolette, dovete digitare nell'ordine: Shift + P, Spazio, Commodore + tasto "-", tasto Crsr Down, 3 volte il tasto Crsr Left, 2 volte il tasto Shift + tasto "Q", tasto Ctrl + 0, tasto Commodore + asterisco (*), chiusura virgolette.

```
0 L=54296:POKE L-4,
  128:POKE L-6,129:
  POKE L-9,9:FOR I=
  1 TO 99:PRINT"[CL
  EAR]" TAB(I)"┐ ▒[
  DOWN][3 LEFT]●●[R
  VS]◥":POKE L,I:NE
  XT
```

11

Simulatore di PRINT AT. Un nuovo esempio per sopperire alle carenze del Basic "ufficiale" Commodore. Assegnando ad "X" un valore compreso tra 0 e 24 ad "Y" un valore compreso tra 0 e 39, si visualizzerà un qualsiasi messaggio (qui rappresentato da un semplice asterisco) sul rigo X a partire dalla posizione Y.

```
1 INPUT "X,Y";X,Y:P
  RINTCHR$(147):POK
  E 783,PEEK(783) A
  ND 254:POKE 781,X
  :POKE 782,Y:SYS65
  520:PRINT"*":GOTO
   1
```

12

Oscillazioni. Uno Sprite oscilla

dall'alto in basso e dal basso in alto simulando, grazie alla funzione sinusoidale, il rimbalzo di una molla. Lo sprite sembra in parte lampeggiare perchè si riferisce ad una zona di memoria in continua attività (!).

```
1 PRINT"[CLEAR]":V=
  53248:POKE V,165:
  POKE V+21,1:POKE
  2040,0:FOR X=0 TO
   198:POKE V+1,141
  +99*SIN(X/9*π):NE
  XT:GOTO 1
```

13

Spritebusters. Divertente effetto di animazione degli otto sprite, utile da attivare nei momenti di pausa dei vostri programmi.

```
1 PRINT"[CLEAR]":V=
  53248:POKE V+32,0
  :POKE V+33,0:POKE
   V+21,255:FOR I=0
   TO 15:POKE V+I,R
  ND(0)*176+55:NEXT
  :GOTO 1
```

14

Grattacieli. New York, Boston, Chicago? Cercate voi stessi di dare un nome al profilo di questa megalopoli.

```
1 PRINT"[CLEAR]":FO
  R V=0 TO 19:A=RND
  (0)*12+3:FOR I=0
  TO A:K=I*40+V:POK
  E 1853-K,64:POKE
  56125-K,1:NEXTI,V
```

15

Parole chiave del Basic. Vedrete apparire le parole chiave del Basic ed i messaggi di errore del vostro computer. Si noti che l'ultimo carattere di ciascun messaggio è sempre in Reverse.

```
1 POKE 53272,23:FOR
   I=0 TO 650:POKE
  1195+I,PEEK(41118
  +I):POKE 55467+I,
  1:NEXT
```

16

Esempio di uso di USR (). Questo breve programma, utile per studiare l'accesso al Linguaggio Macchina (L.M.) da Basic cambierà il colore di schermo ogni volta che si digita RUN (R).

```
1 POKE 785,0:POKE 7
  86,192:I=49153:PO
  KE I-1,238:POKE I
  ,32:POKE I+1,208:
  POKE I+2,96:PRINT
  USR(I)
```

17

Pallina. Una pallina rimbalza sul "pavimento" dello schermo.

```
1 FOR X=0 TO 39:PRI
  NT"[CLEAR]":FOR R
  =0 TO (12+9*SIN(X
  /9*π)):PRINT:NEXT
  :PRINT TAB(X)"o":
  FOR T=0 TO 18:NEX
  TT,X:GOTO 1
```

18

Prontezza di riflessi. Un ennesimo giochino che mette a dura prova i vostri nervi (ma non è poi tanto difficile). La novità consiste nel fatto che è necessario premere il tasto Run/Stop al momento della visualizzazione del messaggio "Premi".

```
1 POKE 808,234:A=RN
  D(0)*9999:FOR T=0
   TO A:NEXT:POKE 8
  08,237:PRINT"[RVS
  ]PREMI":FOR T=0 T
  O 128:NEXTT:GOTO
  1
```

19

Attesa pressione tasto. Con questa semplice riga (da attivare con GOSUB 60000), il vostro Commodore resterà in attesa finchè non sarà premuto un tasto. Subito dopo cancellerà lo schermo.

```
60000 POKE 198,0:WAIT 1
      98,1:PRINTCHR$(14
      7):RETURN
```

20

Falso Ready. Prendete in giro i vostri amici facendo credere loro che il computer si è rotto. Nemmeno premendo Run/Stop e Restore si ritorna alla "normalità". E', infatti, necessario spegnere e riaccendere il computer.

```
1 POKE 808,234:PRIN
  T"READY.":FOR I=0
   TO 1 STEP 0:POKE
   204,0:NEXT
```

21

Cursore Professionale. Questo minilistato, che merita di trovare applicazioni nei vostri programmi, consente di simulare la presenza di un diverso tipo di cursore grazie ad una tecnica molto semplice che suggeriamo di approfondire per vostro conto. Per cambiare il "tipo" di cursore lampeggiante, provate a cambiare il valore 228 presente dopo la POKE 1024+I.

N.B. Attendere alcuni secondi dopo il Run. L'effetto scompare se si cancella lo schermo e se si usa spesso la barra spaziatrice oppure il tasto Del. Se lo schermo così "preparato" viene utilizzato con la funzione INPUT, controllare sempre l'argomento dell'istruzione.

```
1 PRINT"[CLEAR][VER
DE]":POKE 53281,6
:FOR I=0 TO 999:P
OKE 1024+I,228:PO
KE 55296+I,6:NEXT
```

22

Reset. Ecco un comando Basic decisamente utile. Disabilitando il comando LET (di solito poco usato), consente l'utilizzo di RET con l'effetto di simulare la pressione del tasto Run/Stop e Restore. Ogni volta che si usa il comando RET, però, è necessario effettuare la POKE1,54 per utilizzarlo di nuovo.

Se volete "uscire", invece, potete ricorrere a POKE1,55 in qualsiasi momento. La possibilità di ricorrere al trucchetto rimane valida sia che premiate i tasti Run/Stop e Restore, sia che ricorriate (se lo possedete) al tasto di Reset.

```
1 FOR I=40960 TO 49
  151:POKE I,PEEK(I
  ):NEXT:POKE 1,54:
  I=41069:POKE I-81
  ,255:POKE I-80,25
  2:POKE I+81,82
```

23

Reverse Schermo. Tenendo conto del fatto che un carattere in reverse si differenzia dauno normale perchè ha il settimo bit posto a 1, questo semplice For... Next pone in reverse qualsiasi carattere si trovi sullo schermo al momento del Run.

```
1 FOR I=1024 TO 202
  3:POKE I,128 OR P
  EEK(I):NEXT:FOR I
  =0 TO 1 STEP 0:NE
  XT
```

24

Screen color. Ennesimo effetto cromatico che associa, ad ogni pressione di un tasto, un diverso colore dello schermo.

```
1 FOR I=0 TO 1 STEP
   0:P=PEEK(197):PO
  KE 53280,P:NEXT
```

25

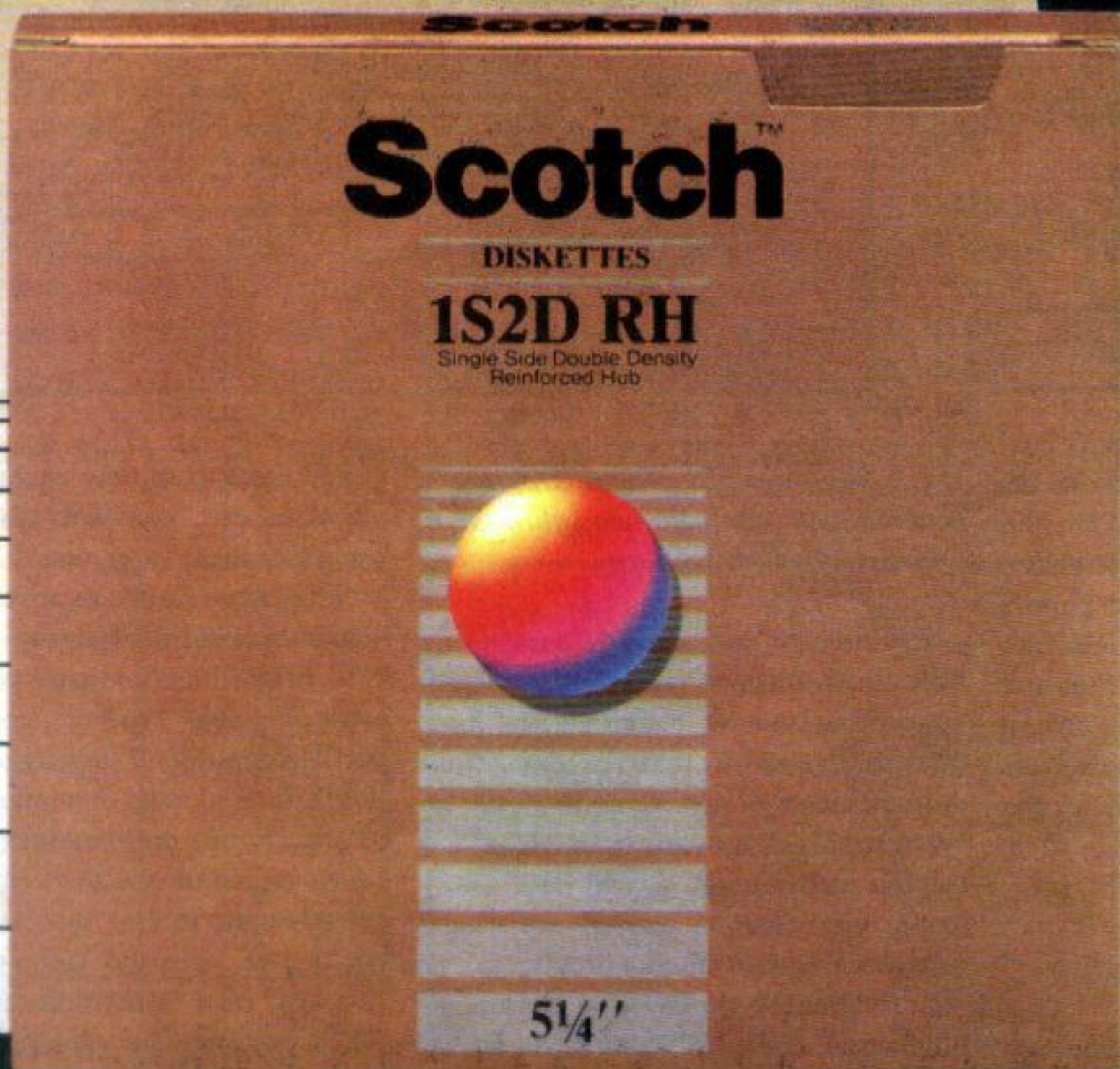
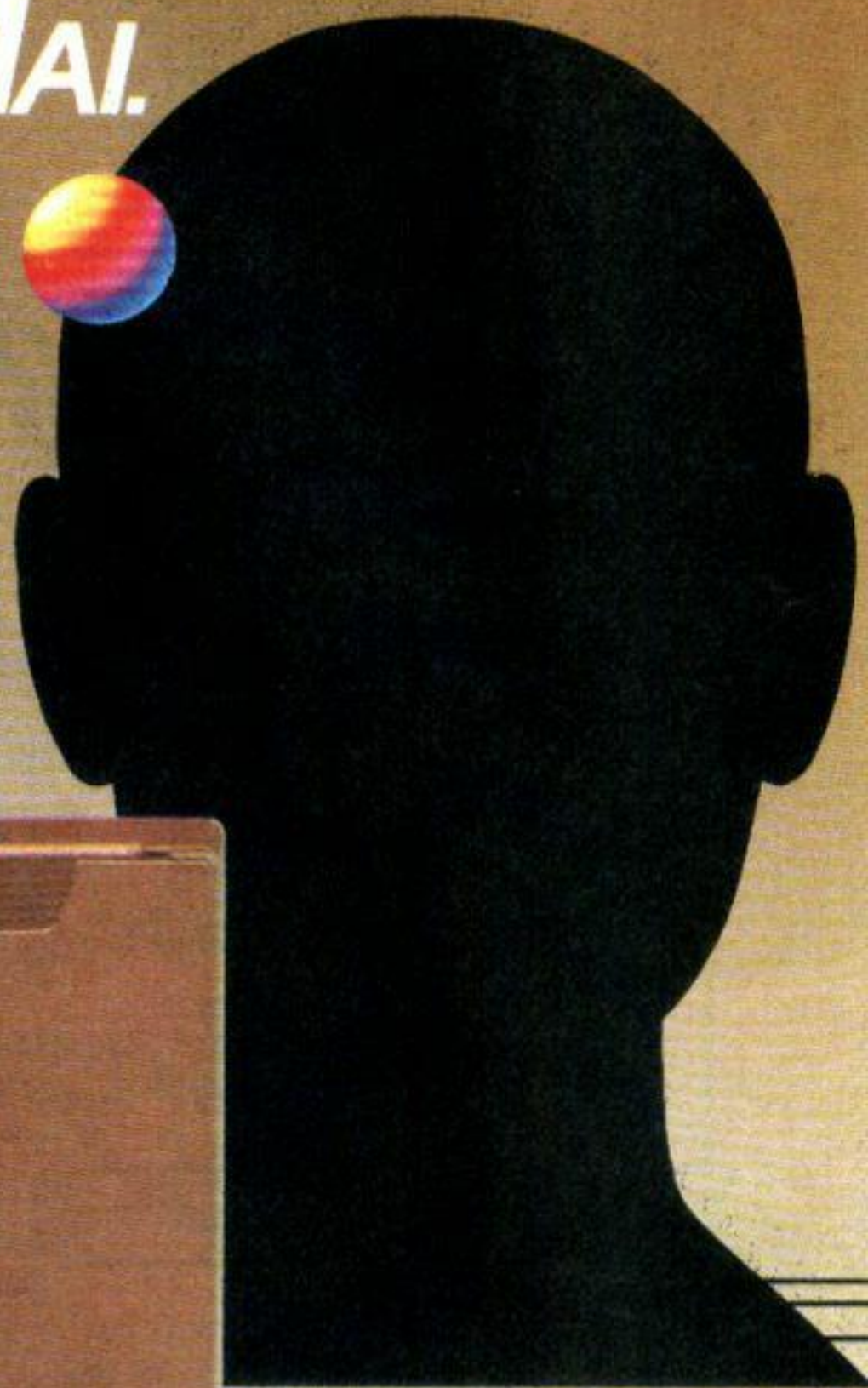
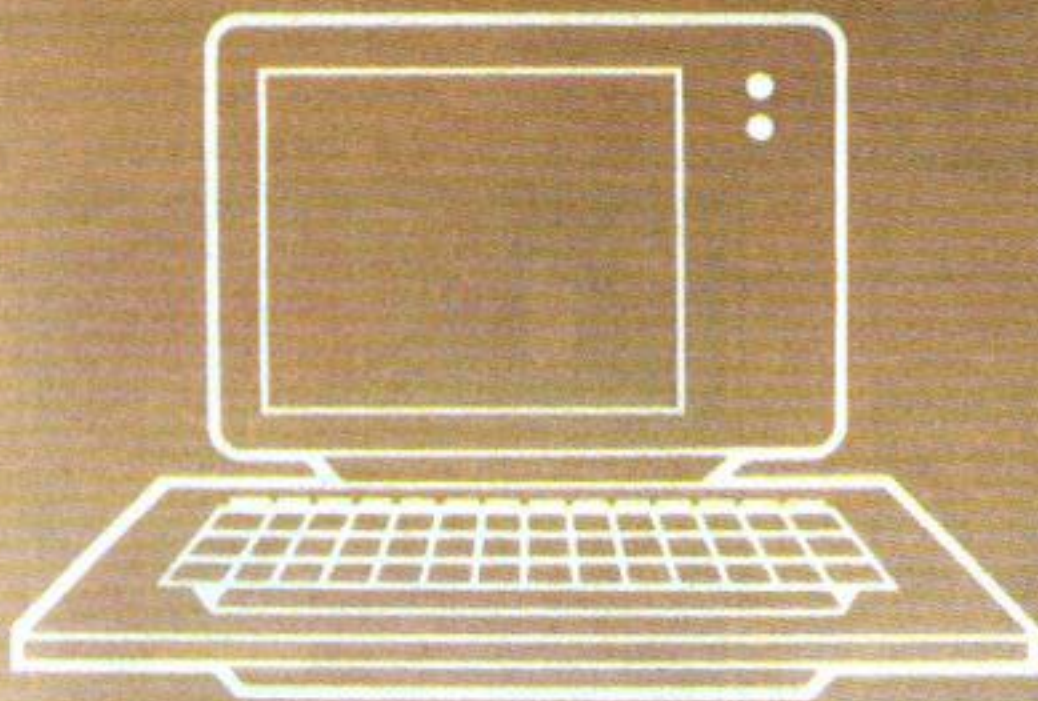
Sprite. Ecco gli otto sprite, tutti in fila, pronti per essere manipolati dall'utente.

```
1 PRINT"[CLEAR][BIA
  NCO]":V=53248:FOR
   I=0 TO 15 STEP 2
  :POKE V+I,31+I*16
  :POKE V+I+1,99:NE
  XT:POKE V+I+5,255
  :POKE V+33,0
```

26

Analisi blocchi. La memoria del Commodore 64 può esser consi-

MEMORIA. PIU' SICURA CHE MAI.

Scotch™
DISKETTES
1S2D RH
Single Side Double Density
Reinforced Hub
5¼''

Scotch è un marchio distribuito nei migliori negozi di audio, video e fotografia con una gamma completa per drive da 5¼'' e da 3½''.

La 3M, leader nella tecnologia dei supporti magnetici, sa quanto sono preziosi i dati che affidi alla memoria del tuo personal computer. Ecco perché ha messo a punto una nuova linea di diskette Scotch che sfruttano la sua ineguagliata esperienza per offrirti un'affidabilità assoluta e una durata tale da consentire di leggere ogni pista milioni e milioni di volte.

Inoltre queste diskette straordinarie concorrono ad assicurare una lunga vita al tuo drive, grazie ad un'abrasività nettamente al di sotto della media.

Scegliendo Scotch, quindi, sei sicuro di scegliere bene.

Scotch™
DISKETTES

DISKETTE SCOTCH: ANNI MEMORIA IN AVANTI.

3M

derata come se fosse formata da 256 blocchi contenenti ciascuno 256 byte (256*256* = 65536). Con questa riga è possibile esaminare, byte per byte, il contenuto di qualsiasi blocco. Per fermare momentaneamente la visualizzazione è necessario premere un tasto qualsiasi e ribatterlo per riprenderla. Ciò può esser fatto per cinque volte. Volendo interrompre la visualizzazione per un numero di volte maggiore, premere il tasto CTRL.

```
1 INPUT "BLOCCO";B:
  IF B>-1 AND B<256
   THEN FOR I=B*256
   TO I+255:PRINTI,
  PEEK(I):WAIT 198,
  1,1:NEXT
```

27

Percentuale. Alla domanda "Val, Perc" rispondete indicando il valore e la percentuale, che a questo si riferisce, separati da una virgola. Verranno visualizzate due informazioni: la percentuale del valore e il montante (capitale + perc.).

```
1 INPUT "VAL,PERC."
  ;V,P:Q=V*P/100:PR
  INT"IL"P"% DI"V"E
  '"Q:PRINTV"+"P"%=
  "Q+V:GOTO 1
```

28

Quotazione in borsa. La prima domanda chiede la quotazione odierna del dollaro. In seguito sarà sufficiente il numero di dollari da cambiare per ottenere il corrispettivo in lire.

```
1 INPUT "QUATAZIONE
   DEL $";L:FOR I=0
   TO 1 STEP 0:INPU
  T "N.$";D:PRINTD"
  $= LIRE"D*L:NEXT
```

29

Sirena. Anche questa riga, come tante altre presentate in precedenza su questa rubrica, crea un effetto sonoro continuato. In questo caso una sirena dal sibilo acuto vi intratterrà piacevolmente durante gli allarmi aerei.

```
1 L=54273:POKE L+23
  ,15:POKE L+5,128:
  POKE L+3,33:FOR I
  =0 TO 199:POKE L,
  99+49*SIN(I/99*π)
  :NEXT:GOTO 1
```

30

Caratteri. Tutti i caratteri del Commodore 64 messi in bell'ordine al centro dello schermo.

```
1 FOR V=0 TO 7:FOR
  I=0 TO 31:POKE 10
  28+I+V*40,V*32+I:
  POKE 55300+I+V*40
  ,1:NEXTI,V
```

Una riga per C-16 Plus/4

Ed ecco una manciata di righe specifiche per il Commodore 16 e Plus/4 ideate dal nostro Michele Maggi.

Treno. Un simpatico rumore di treno in corsa generato dal vostro computer.

```
1 VOL8:SOUND3,500,3:
  FORJ=1TO2:FORI=8TO9:
  SOUND3,I*100,3:
  NEXTI,J:GOTO1
```

Effetto sonoro. Un interessante effetto sonoro da utilizzare nei vostri programmi.

```
1 VOL8:FORI=1TO1E3STEP40:
  SOUND1,50,5:SOUND2,I,5:
  NEXT:FORI=1E3TO1STEP-50
  :SOUND1,99,5:SOUND2,I,5
  :NEXT:RUN
```

Reset. La SYS di reset del vostro Commodore 16 da usare con parsimonia e prudenza...

```
1 PRINT"SE PREMI UN TASTO":
  PRINT"RESETTI IL TUO C-16":
  GETKEYA$:SYS65526
```

Schermo pulsante. Un effetto "pulsante" dallo schermo ottenuto ricorrendo a tutti i colori in tutte le sfumature possibili.

```
1 FORJ=1TO9:
  FORI=0TO7STEP.1:
  COLOR4,J,I:
  COLOR0,J,I:NEXT:
  FORI=7TO0STEP-.1:
  COLOR0,J,I:
  COLOR4,J,I:
  NEXTI,J:RUN
```

Bordo lampeggiante. Un lampeggio sul bordo dello schermo del C-16. Ideale per richiamare l'attenzione durante la pausa di un programma.

```
1 FORI=1TO16:
  COLOR4,I,5:NEXT:
  GOTO1
```

Nota Bene

Alcune righe tra quelle pubblicate sembrano possedere più di 80 caratteri e, come tali, inaccettabili dal computer.

Nei casi in cui ci si accorga che la riga è troppo lunga, è necessario ricorrere alle abbreviazioni dei comandi così come indicato nell'appendice specifica riportata nel manuale del computer in vostro possesso.

Ad esempio invece di scrivere PRINT è possibile abbreviare col punto interrogativo (?). Invece di POKE potete scrivere il carattere "P" seguito dal carattere che viene visualizzato premendo contemporaneamente il tasto shift insieme con "O". Tutte le abbreviazioni possibili, lo ripetiamo, sono riportate in una delle appendici di qualsiasi manuale Commodore.

Nel caso sbagliate a digitare i microlistati che superano, in lunghezza, gli ottanta caratteri (SYNTAX ERROR o altri tipi di errore), è necessario, per sicurezza, ribatterli per intero e non apportare modifiche alla riga visualizzata con l'istruzione LIST.

La partecipazione dei lettori è gradita e compensata, in caso di pubblicazione, con materiale della Systems editoriale (libri, fascicoli arretrati, abbonamenti, eccetera).

Inviate le vostre 1 RIGA, purchè nella misura di almeno dieci per volta. Non ci è infatti possibile raggruppare i (moltissimi) micro-programmi che pervengono singolarmente in Redazione.

Inviate i vostri lavori su carta, (meglio se su nastro) corredati, ciascuno, di una breve spiegazione sulla funzione che compiono, proprio come li vedete pubblicati in queste pagine.

Ricordate di indicare chiaramente nome, cognome, via e città, e indirizzate a:

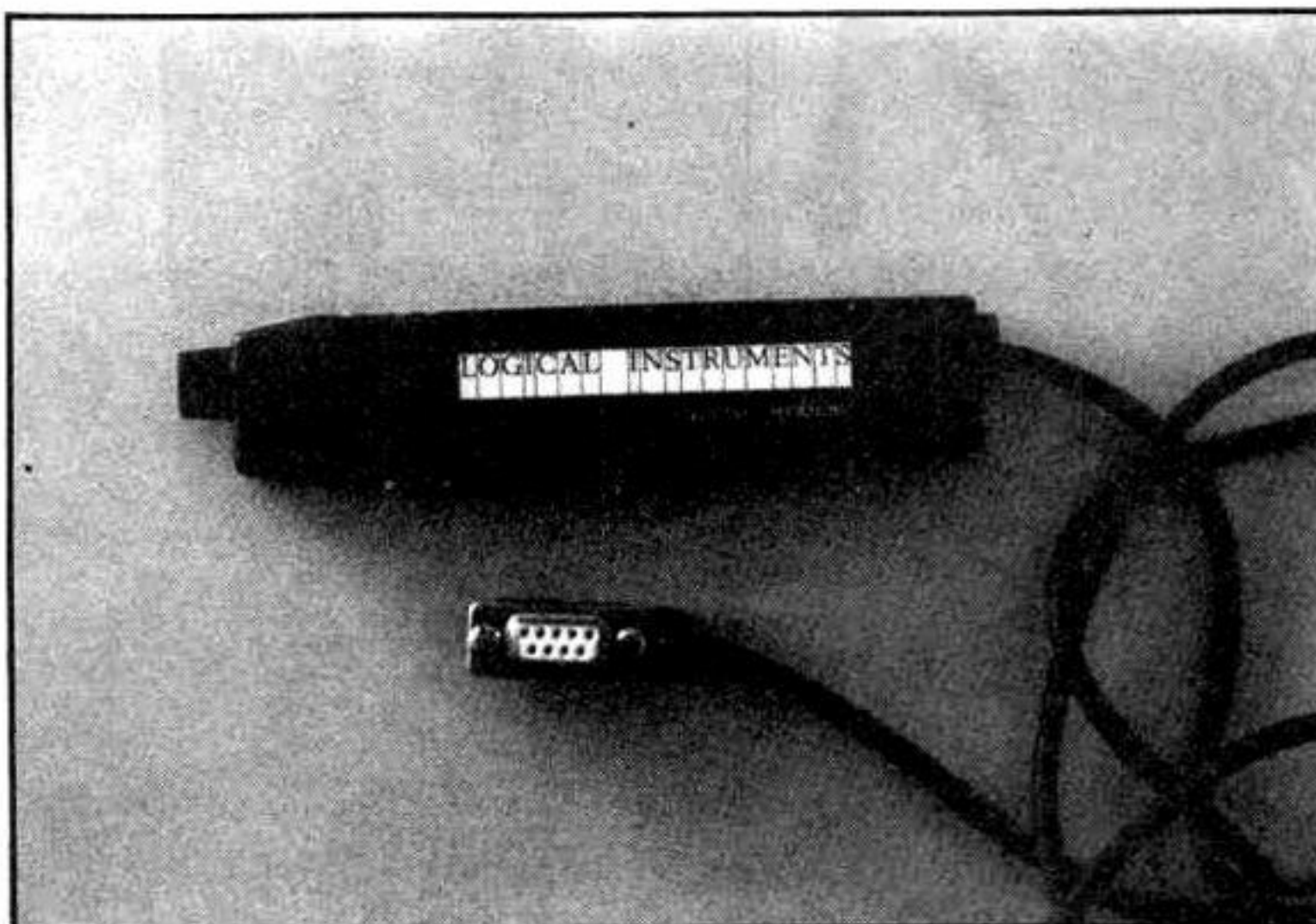
Commodore Computer Club
SYSTEMS EDITORIALE

Rubrica "1 Riga"

Viale Famagosta, 75
20142 MILANO

Penna ottica per il C-64

Come realizzare (o procurarsi già pronto per l'uso) un semplice accessorio utile in più di un'occasione.

di Andrea e Alberto Boriani

Continua con questo articolo la serie di progetti hardware utili non solo per aumentare le proprie conoscenze di elettronica, ma anche per dotare il vostro sistema di accessori decisamente economici.

Finora infatti, sulle pagine di questa rivista ci eravamo occupati di solo software: quel "qualcosa", cioè, di poco tangibile che va inserito volta per volta nel calcolatore attraverso la tastiera, il registratore o i dischi.

Finalmente ora cominceremo anche ad occuparci del corpo, e non solo della mente, del vostro amico elettronico.

Premettiamo che, in questa serie di articoli, non si chiederà di smontare il computer per vedere come è fatto dentro, bensì ci occuperemo di procurare al C—64 (ed ai suoi "parenti") una serie di accessori utili e, talvolta, indispensabili. Di volta in volta possiamo limitarci ad imparare come funzionano oppure, addirittura, a costruirli.

Nel presente articolo, in particolare, ci occuperemo del progetto e della realizzazione di una penna ottica o light pen che dir si voglia.

Probabilmente qualcuno di voi avrà già sentito parlare della penna ottica e, magari, avrà avuto modo anche di utilizzarla su computer più grandi.

Per chi invece sentisse questo nome per la prima volta o, comunque, volesse saperne di più, diremo che questo strumento, molto usato nel campo della "computer grafica", ha l'aspetto di una vera e propria penna per scrivere che risulta però collegata al computer attraverso un cavo elettrico.

La penna ottica, naturalmente, assomiglia solo esternamente ad una penna biro.

Internamente, infatti, è dotata di un particolare circuito elettronico che la mette in grado di "scrivere" sullo schermo del calcolatore.

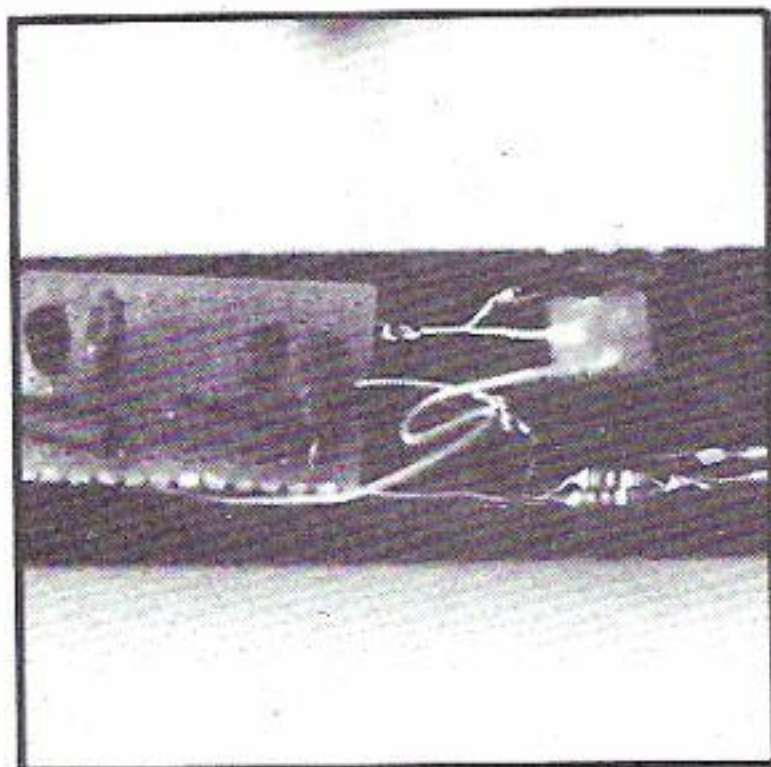
In realtà vedremo che la penna ottica si limita a comunicare al computer, istante per istante, le coordinate di schermo in cui si trova la punta della penna stessa, ed è effettivamente il computer che, attraverso un apposito programma, si occupa di accendere il punto luminoso corrispondente.

Vi renderete perfettamente conto delle ampie possibilità di utilizzo di un apparecchio del genere, sia in campo grafico, per disegnare direttamente sullo schermo, sia in campi diversi, come ad esempio per fare una scelta all'interno di un menu senza usare la tastiera, ma semplicemente indicando con la penna ottica la vostra decisione.

Come funziona una light pen

Passiamo ora a vedere in modo più specifico come funziona una penna ottica.

Prima di tutto parleremo del funzionamento di un televisore o di un monitor televisivo.

Come forse già saprete, la luminosità prodotta dallo schermo del computer ha ben poco a che fare con quella, ad esempio, di una comune lampadina.

La differenza fondamentale sta nel fatto che mentre una lampadina diffonde luce in tutte le direzioni, la luce prodotta all'interno del vostro schermo è in realtà dovuta al movimento rapidissimo di un unico raggio sottilissimo, inviato con molta precisione su ogni punto dello schermo .

Immaginate di avere in mano un pennarello e di voler colorare in modo uniforme un foglio di carta; il metodo più razionale per farlo è di tracciare molte righe parallele tra loro, una vicina all'altra, a partire dalla cima del foglio fino al fondo. In questo modo avrete realizzato una "scansione" del tutto simile a quella operata dal pennellino elettrico che, all'interno di un qualsiasi televisore, invia un raggio di elettroni sullo schermo.

A seconda, poi, che gli elettroni arrivino effettivamente sullo schermo o vengano fermati prima, si formeranno immagini differenti come, ad esempio, i caratteri alfanumerici.

Vi potete render conto di questa realtà osservando molto da vicino una messaggio o un disegno del computer: scritte e disegni, infatti, appariranno come composti da più righe orizzontali prodotte, appunto, dalla scansione video.

Il vostro computer possiede al suo interno un registro di memoria contenente sempre la posizione del "pennellino elettronico" che continuamente scandisce lo schermo. Se, per assurdo, potessimo seguire con gli occhi il movimento del pennello e premere un tasto tutte le volte che il punto luminoso si trova nella posizione che interessa, potremmo comunicare al computer il punto preciso in cui tracciare un punto, una riga, un cerchio o qualsiasi altra cosa.

Naturalmente un occhio umano non è in grado di fare una cosa del genere perchè il pennellino elettronico si muove ad una velocità tale da riempire lo schermo televisivo con 625 righe per ben 25 volte al secondo.

Ma dove un occhio umano è insufficiente può intervenire un occhio elettronico.

All'interno di una penna ottica si trova, infatti, un piccolo occhio elettronico chiamato "fotodiodo". Questo è notevolmente più veloce di un occhio umano, tanto da potergli affidare il compito di riferire al computer la posizione del pennellino sullo schermo.

Al fotodiodo, contenuto nella penna ottica, è infatti collegato un circuito transistorizzato che si occupa di inviare i segnali opportuni al computer ogni volta che il fotodiodo "vede" la luce proveniente dal pennello.

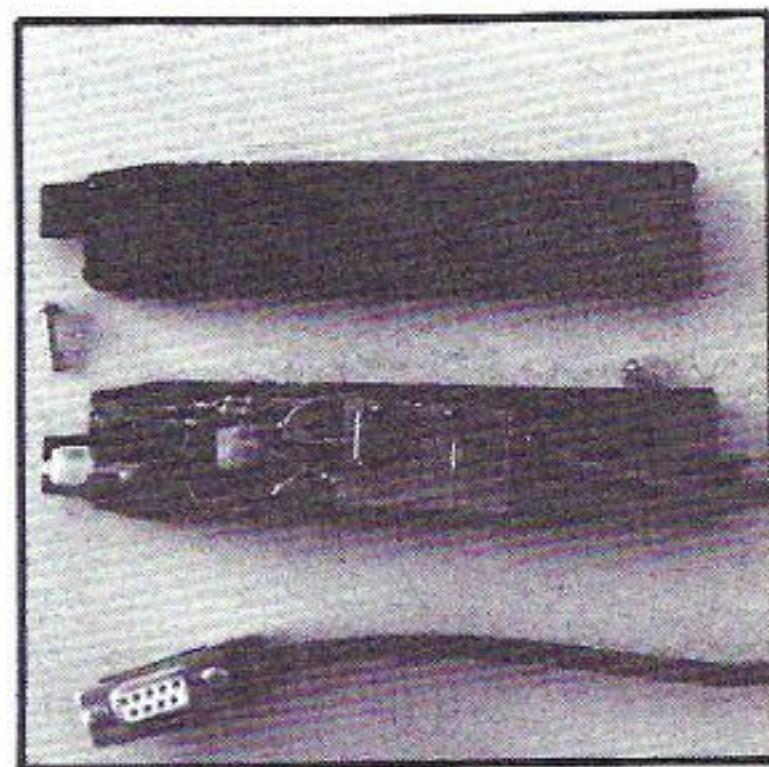
Lo schermo elettrico

Non è questa la sede per dilungarsi in spiegazioni specifiche sul modo in cui si è giunti al progetto presentato in questo articolo: ci limiteremo a dire che il fotodiodo si comporta, grossomodo, come un interruttore che si chiude tutte le volte che vede luce e si apre quando vede buio; il segnale prodotto dal fotodiodo viene poi amplificato e "ripulito" dai due transistor e dal condensatore presenti nel circuito in modo da esser letto senza problemi dal computer.

La penna ottica è dotata anche di un pulsante che si rende utile per molte applicazioni.

Realizzazione del circuito

Chiunque desiderasse realizzare la penna ottica, troverà in queste pagine lo schema elettrico dell'accessorio e, ovviamente, lo schema del suo collegamento al C—64 oltre alla lista dei componenti.

Data la semplicità del circuito, diremo

solo che questo può essere montato sia punto per punto, sia su una basetta "millefori" da chiunque abbia un minimo di esperienza in montaggi elettronici. Per chi desiderasse invece un montaggio più pulito e professionale potrà richiedere i componenti in scatola di montaggio completa di contenitore, cavo e connettore. La confezione, oltre all'indubbio vantaggio di avere a disposizione, subito, tutto il materiale occorrente, contiene al suo interno, soprattutto, l'apposito circuito stampato che riduce a zero i pericoli di mancato funzionamento a causa di errori di montaggio.

Per i più pigri (oppure per chi non si sente in grado di realizzare la penna elettronica) è anche disponibile la versione già montata e collaudata dell'accessorio descritto.

Il collegamento della penna ottica al computer avviene attraverso la control port N.1 a cui si collega di solito il joystick.

Le coordinate della penna ottica possono essere lette nelle locazioni 53267 e 53268:

```
X=PEEK(53267)
Y=PEEK(53268)
```

Il pulsante della penna ottica può essere letto tramite:

```
A=(PEEK(56321)AND1)
```

La variabile A assumerà il valore 0 se il tasto è premuto, 1 in caso contrario.

Applicazioni

Per facilitare il collaudo della penna ottica, è riportato il listato di un pro-

gramma Basic per il Commodore 64 in grado di mostrare il buon funzionamento dello strumento.

Il programma sfrutta le ben note routines grafiche di Danilo Toma (seconda versione N.14 Commodore Computer Club) che, come al solito, dovranno essere caricate e lanciate prima del programma specifico.

Come in tutti i programmi che contengono l'istruzione POKE è buona norma salvare il programma subito dopo averlo digitato e PRIMA di dare il RUN, in modo da mettersi al riparo da eventuali errori di battitura che potrebbero causare un inchiodamento del calcolatore con la necessità quindi di dover ribattere tutto da capo.

Funzionamento del programma

Questo programma permette di disegnare sullo schermo a mano libera, di tracciare linee (dati gli estremi), e di tracciare cerchi (dati il centro e un punto sulla circonferenza).

Prima di accendere il computer (mi raccomando: PRIMA) inserite il connettore della penna ottica nella porta 1, poi accendete il computer, caricate le routines grafiche, date il run e infine caricate il programma.

Dopo il RUN lo schermo si colora di bianco ed appare una crocetta che potrete spostare semplicemente accostando la penna ottica allo schermo e muovendola come per disegnare. Se la crocetta non si sposta, provate ad alzare il livello di luminosità o di contrasto del televisore.

A questo punto avete a disposizione i seguenti comandi:

Tasto 1: Si seleziona la scrittura a mano libera. Premendo il pulsante sulla penna ottica, e muovendola sullo schermo, oltre a muovere la crocetta lascerete anche una traccia nera mentre rilasciando il pulsante sposterete solo la crocetta.

Tasto 2: Si disabilita la funzione precedente: anche premendo il tasto sulla penna ottica non scriverete nulla.

Tasto 3: Si abilita la funzione "tracciamento linee". Prima di premere questo tasto posizionate con la penna ottica la crocetta sul punto in cui volete un estremo della riga. Allontanate quindi la penna dallo schermo e premete il tasto 3. Vedrete apparire un puntino nel centro della crocetta. Riavvicinate la penna allo schermo e procedete nella stessa maniera di prima. Dopo la nuova pressione del tasto 3 apparirà una linea congiungente il punto precedentemente fissato con il centro della crocetta.

Tasto 4: Si abilita la funzione di tracciamento cerchi. Come per la funzione tracciamento linee dovrete, prima di premere il tasto 4, spostare la crocetta nel punto dove volete il centro della circonferenza. Premuto 4 apparirà un puntino indicante il centro. Potrete quindi spostare la crocetta e, premendo il tasto 4 una seconda volta, tracciare la circonferenza di raggio pari alla distanza tra il puntino precedentemente tracciato e la posizione della crocetta.

Tasto 0: Permette la cancellazione di tutto lo schermo grafico.

Come avrete notato non è stata implementata la funzione di cancellazione parziale. A voi quindi il compito di realizzare la routine per usare la penna ottica anche come "gomma ottica".

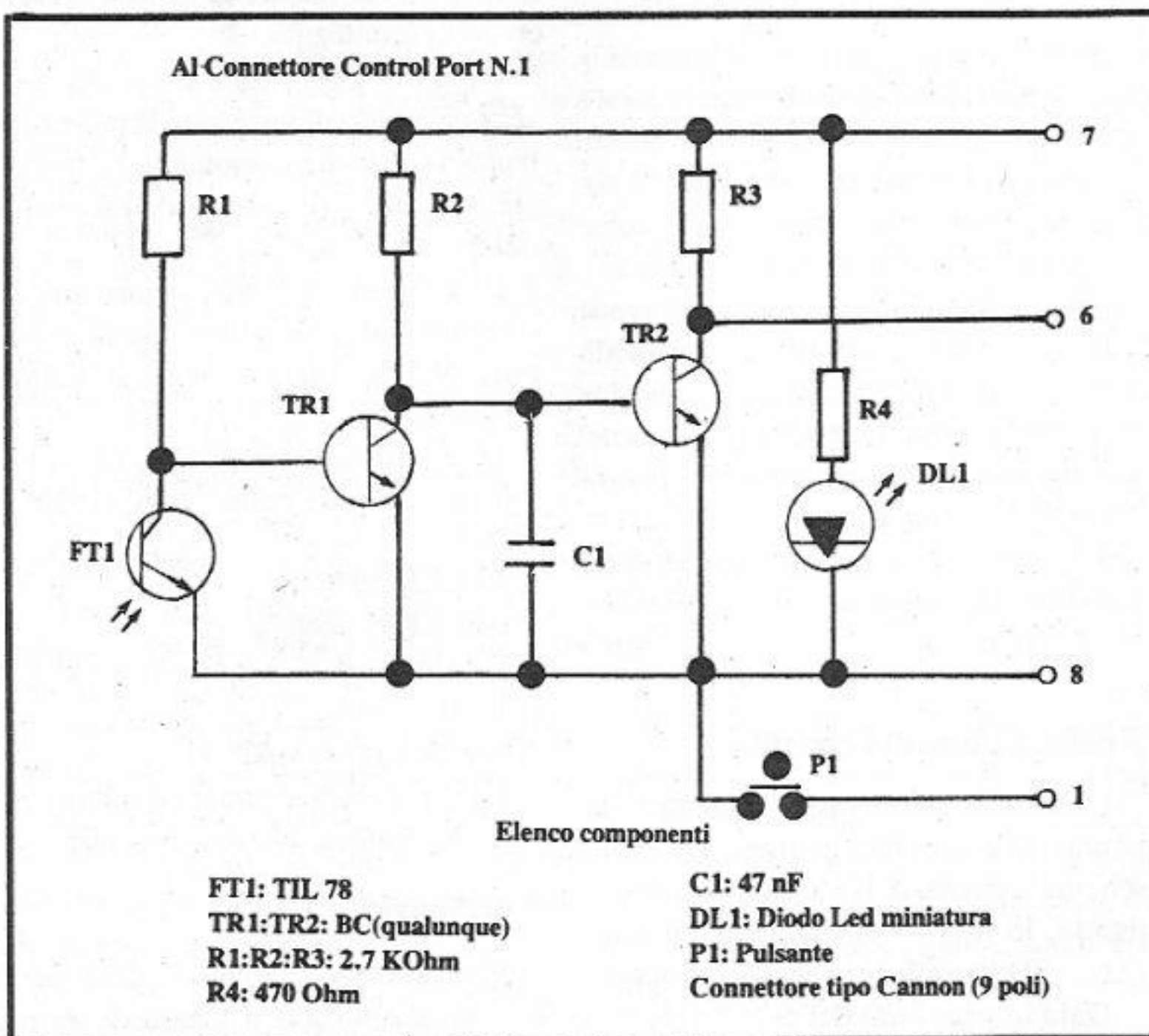
Noterete, soprattutto durante il disegno a mano libera, una leggera instabilità (ondeggiamento) della crocetta sull'asse orizzontale, causata da imprecisioni di lettura del computer, non dalla penna ottica.

Ricordatevi, quando sviluppate programmi che utilizzano la penna ottica che l' "occhio elettronico" in essa contenuto è un po'... daltonico, ovvero non vede allo stesso modo tutti i colori. Se possibile utilizzate sempre il bianco, o comunque colori molto chiari, come colore di sfondo.

Come detto sopra, questa penna ottica può essere acquistata per corrispondenza scrivendo a:

Logical Instruments
Via Torricelli 3
Bollate (Milano).

Il costo è di L. 30.000 per la versione in scatola di montaggio e L. 35.000 per la versione montata e collaudata, più le spese di spedizione (pagamento al postino).

Potete anche ordinare telefonicamente allo 02/3511871, lasciando il vostro nome e indirizzo alla segreteria telefonica.

```
  5 F=-1
 10 ←CLEAR:←GRAF1,0:←COL OR 1:P
    OKE 53281,1
 20 V=53248:A=50496
 21 FOR X=A+832 TO A+832+64:POK
    E X,0:NEXT
 22 FOR X=A+833 TO A+833+20 STE
    P 3:POKE X,8:NEXT
 23 POKE A+862,255:POKE A+863,0
    :POKE A+864,255
 24 FOR X=A+875 TO A+872+20 STE
    P 3:POKE X,8:NEXT
 30 POKE V,50:POKE V+1,70:POKE
    V+16,0:POKE V+28,0:POKE 532
    40,34
 40 POKE V+27,0:POKE V+23,0:POK
    E V+29,0:POKE V+21,1:POKE V
    +39,0
 50 X=PEEK(V+19)*2-160:Y=PEEK(V
    +20)-7:IF Y<0 THEN 70
 60 POKE V+1,Y:IF X<0 THEN 70
 61 IF X<=255 THEN POKE V,X:POK
    E V+16,0:GOTO 70
 62 POKE V,X-255:POKE V+16,1
 70 X1=X-170:Y1=150-Y
 80 GET A$
 85 IF H=2 THEN F=-1
 90 IF A$="1" THEN H=1:X2=X1:Y2
    =Y1
100 IF A$="2" THEN H=2
150 IF H=1 AND (PEEK(56321) AND
     1)=1 THEN Y2=Y1:X2=X1
200 IF H=1 THEN IF (PEEK(56321)
     AND 1)=0 THEN :←DRAWX2,Y2,
    0,X1,Y1,0:X2=X1:Y2=Y1
210 IF A$="0" THEN H=2:←CLEAR
220 IF A$="3" THEN H=3:F=F+1:←P
    LOTX1,Y1,0:GOTO 230
225 GOTO 250
230 IF F/2=INT(F/2) THEN X2=X1:
    Y2=Y1:GOTO 250
240 ←DRAWX2,Y2,0,X1,Y1,0
250 IF A$="4" THEN H=4:F=F+1:←P
    LOTX1,Y1,0:GOTO 270
260 GOTO 290
270 IF F/2=INT(F/2) THEN X2=X1:
    Y2=Y1:GOTO 290
275 R=SQR((X2-X1)↑2+(Y2-Y1)↑2)
280 ←CIRCLEX2,Y2,0,R,R:←COL OR
    0:←PLOTX2,Y2,0:←COL OR 1
290 REM
300 GOTO 50
```

Come disabilitare il tasto di Reset

Come evitare che un vostro programma possa esser fermato anche ricorrendo al pulsante di Reset.

di Antonio Visconti

In un recente articolo, pubblicato sul n. 28, abbiamo spiegato in che modo costruire un pulsante di Reset da collegare al Commodore 64, che ne è privo.

Un nostro lettore ci ha fatto notare che alcuni programmi, interrotti ricorrendo al pulsante, rischiano di essere sprotetti dato che, una volta interrotta l'esecuzione, è agevole, per chi se ne intende, eliminare le eventuali protezioni ivi inserite.

In questo articolo viene illustrato il modo di rendere inefficace il tasto di Reset e a tal fine sono riportate e commentate alcune routine in Linguaggio Macchina (L.M.) del Sistema Operativo (S.O.).

Inoltre viene presentato un programma Basic che, aggiunto ad un qualsiasi altro programma, scritto anch'esso in Basic, "disabilita" il Reset nel senso che una sua pressione ha, come unico effetto, quello di generare il comando RUN.

Lo stesso effetto si ottiene anche premendo i tasti Run/Stop e Restore e questo particolare rende interessante l'articolo anche per coloro che, non possedendo il tasto di Reset, desiderano egualmente apprendere nuove tecniche di protezione.

Nonostante ciò che abbiamo detto, non sono richieste conoscenze di L.M. per l'utilizzo del programma, che sono invece indispensabili per la comprensione del "come funziona".

Chi ha fretta di sperimentare questa nuova tecnica, può dunque evitare la lettura dei prossimi due paragrafi, e "saltare" direttamente all'ultimo (Come utilizzare il programma).

Come funziona

Tutti coloro che si sono procurati un tasto di Reset, si sono anche imbattuti in programmi che non vengono disabilitati: l'unica soluzione è lo spegnimento del computer.

Questo strano comportamento dell'elaboratore è stato, per molti, una sorpresa ed anche una grossa delusione. Lo spegnimento e la riaccensione costringe, purtroppo (!), a ricaricare il programma senza avere la possibilità di esaminarlo o, peggio, di copiarlo.

Per la delusione non possiamo fare niente, ma per la sorpresa possiamo dir qualcosa.

Quando viene premuto il ta-

sto di Reset, un particolare piedino del microprocessore viene collegato con la massa. Questo avvenimento rappresenta un segnale che lo costringe ad abbandonare tutte le operazioni in corso e ad eseguire una particolare routine, la cosiddetta routine di Reset, che può essere richiamata normalmente anche digitando SYS 64738.

Quindi, per capire come funziona, bisogna dare uno sguardo al disassemblato di tale routine, che è riportato in queste pagine nella sola parte che interessa.

Come si può notare, la prima operazione che viene eseguita è il salto ad una routine che esegue il test per verificare la presenza di una cartridge nell'apposito connettore. Se questa viene "riconosciuta", viene causato un salto ad un programma il cui indirizzo di partenza si trova in $8000 (decimale = 32768).

Nella zona individuata dall'indirizzo 32768 a 40959 è presente, di norma, una zona di memoria RAM. Anche le cartucce che si inseriscono nell'apposito connettore posseggono, al loro interno, un circuito integrato ROM (oppure Eprom o Prom) con gli stessi indirizzi. Non appena si accende il computer, però, la zona RAM non viene più "riconosciuta" ed il C-64 ritiene che, nella suddetta zona, sia presente soltanto memoria a sola lettura. Naturalmente le cartucce che sono attivate non appena si accende il computer (praticamente tutte) posseggono una sequenza di caratteri particolari che rappresentano, appunto, il codice di partenza.

Poichè abbiamo detto che il controllo viene effettuato solo su alcuni particolari byte e dato che questi, di norma, sono locazioni RAM, possiamo mettere in tali locazioni l'indirizzo di una nostra routine. Per far questo dobbiamo soltanto simulare la presenza di una cartridge.

La routine che esegue il test ci permetterà di "ingannare" il sistema operativo facendogli credere di trovare una cartridge nell' apposito alloggiamento.

Il test avviene in maniera molto semplice: viene confrontato il contenuto di 5 locazioni di memoria RAM (da$8004 a $8008) con altrettanti byte ROM (la tabella che va da $FD10 a FD14). Se i cinque valori coincidono, il sistema operativo si convince che sia inserita una cartridge. Quindi basta ricopiare il contenuto della tabella nelle suddette locazioni che il gioco è fatto.

Il programma

Ci proponiamo, a questo punto, di rendere innoffensivo il tasto di Reset provocando, ad ogni tentativo di pressione, un nuovo RUN.

Altrove è riportato il disassemblato della nostra routine.

Quando viene eseguita, il buffer di tastiera viene riempito con i caratteri "R", "U", "N", e CHR$(13); in seguito viene effettuato un salto alla routine di "warm start" che mette a posto alcuni puntatori e, quello che più interessa "scarica" il buffer di tastiera.

Il programma Basic pubblicato carica in memoria la routine e compie tutte le operazioni necessarie all'abilitazione: copia la tabella, gli indirizzi e salta a $801B. Il Top della memoria viene posto a $8000 per proteggere la nostra routine e la tabella. Questo comporta una diminuzione di 8K della memoria disponibile.

Come utilizzare il programma

Copiate il listato e salvatelo su disco o nastro. Per vedere come funziona basta dare il RUN al semplice Demo pubblicato.

Quest'ultimo, che ha la sola funzione dimostrativa, ha il semplice compito di visualizzare di continuo l'orologio interno del C–64. E' ovvio che il lettore potrà aggiungere la semplice protezione a qualsiasi programma Basic in suo possesso.

ATTENZIONE: l'unico modo per disabilitare il programma è lo spegnimento del computer. Salvate quindi il listato prima di qualsiasi prova.

Per utilizzarlo in un vostro programma cancellate le linee da 10 a 99, ed iniziate con un GOSUB 60000.

Combinato con un sistema di autostart, la disabilitazione del Reset è già una buona forma di protezione, che impedirà ad occhi indiscreti di dare uno sguardo ai vostri listati.

```
Routine di Reset

FCE2  A6 FF     LDX $FF        X=255
FCE4  78        SEI            Disab. IRQ
FCE5  9A        TXS
FCE6  D8        CLD
FCE7  20 02 FD  JSR $FD02      Test cartridge
FCEA  D0 03     BNE $FCEF      Se trovata...
FCEC  6C 00 80  JMP ($8000)    ...esegui routine

Esame per la presenza di una cartridge

FD02  A6 05     LDX $05          Lunghezza=5
FD04  BD 0F FD  LDA $FD10-1,X    Compara i byte...
FD07  DD 03 80  CMP $8004-1,X  ...con quelli in ROM
FD0A  D0 03     BNE $FD0F        Uscita se negativo
FD0C  CA        DEX              Ripeti...
FD0D  D0 F5     BNE $FD04        ...cinque volte
FD0F  60        RTS              X=0 se esame positivo

Disabilita reset

8000  09 08                    Indirizzo routine
8002  09 08                    di disabilitazione.
8004  $C3 $C2 $CD $38 $30      Caratteri tabella

8009  A2 04     LDX #$04        4 caratteri...
800B  BD 28 80  LDA $8029-1,X   ..nel buffer..
800E  9D 76 02  STA $0277-1,X   ..di tastiera
8011  CA        DEX
8012  D0 F7     BNE $800B
8014  A9 04     LDA #$04
8016  85 C6     STA $C6
8018  4C 69 FE  JMP $FE69 Salta al warm start

801B  A9 00     LDA #$00 Fine memoria a 32768
801D  85 37     STA $37
801F  A5 80     LDA $80
8021  85 38     STA $38
8023  A9 EF     LDA #$EF Disabilita Run/Stop
8025  8D 28 03  STA $0328
8028  60        RTS

8029  52 55 4E 0D          RUN+CHR$(13)
```

```
15 REM  DISABILITA TASTO RESET
16 :
17 REM  SOLO PER COMMODORE 64
20 :
22 REM  BY ANTONIO VISCONTI
25 :
35 GOSUB 60000:REM  APPLICAZIO
   NE DELLA PROTEZIONE
45 TI$="000000":PRINTCHR$(147)
47 PRINT:PRINT"NON PUOI INTERR
   OMPERE IL PROGRAMMA"
48 PRINT"NE' CON RUN/STOP E RE
   STORE"
49 PRINT"NE' PREMENDO IL TASTO
    RESET"
50 PRINTCHR$(19),TI$:GOTO 50
96 :
98 :
99 :
59998 REM  SUBROUTINE DI PROTEZIO
      NE
59999 REM  DA USARE UNA SOLA VOLT
      A
60000 POKE 808,234:FOR I=0 TO 44:
      READ Q:C=C+Q:POKE 32768+I,Q
      :NEXTI
60010 IF C<>5153 THEN PRINT"ERROR
      E NEI DATA":STOP
60020 SYS32795:RETURN
60030 DATA 9,128,9,128,195,194,20
      5,56,48,162
60040 DATA 4,189,40,128,157,118,2
      ,202,208,247
60050 DATA 169,4,133,198,76,105,2
      54,169,0,133
60060 DATA 55,165,128,133,56,169,
      239,141,40,3
60070 DATA 96,82,85,78,13
```

ALLARME ROSSO

(per i principianti)

Nel digitare righe di programma basic che contengono istruzioni DATA, è piuttosto facile incorrere in errori di digitazione. Supponiamo che un'ipotetica linea basic numerata con 1200 debba contenere i tre valori: 123, 456, 789. Ecco alcuni esempi di errori più frequentemente commessi:

1200 DATA,123,456,789

C'è una virgola dopo la parola "DATA". I dati letti dal computer sono, in questo caso, quattro: 0, 123, 456, 789. Se, infatti, non figura alcun carattere dopo l'istruzione DATA, automaticamente viene assunto il valore nullo (0).

1200 DATA 123,456,789,

In questo caso, dopo il numero 789, il computer, grazie alla presenza della virgola erroneamente inserita, "crede" che ci sia un altro valore e, non trovandolo, lo assume come nullo (0).

1200 DATA 1234,56,789

La virgola è posizionata male, vale a dire dopo il carattere "4" e non dopo il carattere "3". Il computer non può sapere se il valore esatto è 123 oppure 1234 e individuare un errore, in questo caso, risulta piuttosto laborioso.

Accetti la sfida?

Una nuova iniziativa per stimolare la fantasia dei lettori

Il notevole, inaspettato, successo avuto con l'articolo sulle "Curve di Peano" ci ha indotto a sfidare nuovamente i lettori proponendo la stesura di un programma di grafica. L'idea è nata esaminando i meravigliosi disegni che possono essere riprodotti in modo semplice grazie ad una penna e ad alcune ruote e corone dentate.

Come potete vedere in questa pagina, i disegni vengono tracciati in virtù di una tecnica che ricorre al tracciamento di particolari curve evolventi.

Gli elementi che consentono di generare le infinite varietà di curve sono soltanto tre:

- Diametro della "ruota" forata.
- Distanza dal centro del foro in cui si inserisce la punta della penna.
- Diametro della circonferenza all'interno (ma potrebbe essere anche all'esterno) della quale gira la ruota.

Supponendo che la ruota giri senza strisciare non dovrebbe esser difficile risalire all'algoritmo che, tenendo conto dei tre fattori indicati, consente di tracciare in alta risoluzione una infinita varietà di curve.

Che cosa dovete fare?

Verranno presi in considerazione, premiati e, se meritevoli, pubblicati, i listati Basic che avranno le seguenti caratteristiche:

- Brevità. Meno righe Basic richiederà il listato, più probabilità avrà di essere apprezzato.
- Universalità. Saranno privilegiati i lettori che invieranno diverse versioni in modo da realizzare le curve su C64, C16, C128, Vic20.

Se possedete un solo computer, poco male: partecipate egualmente alla sfida.

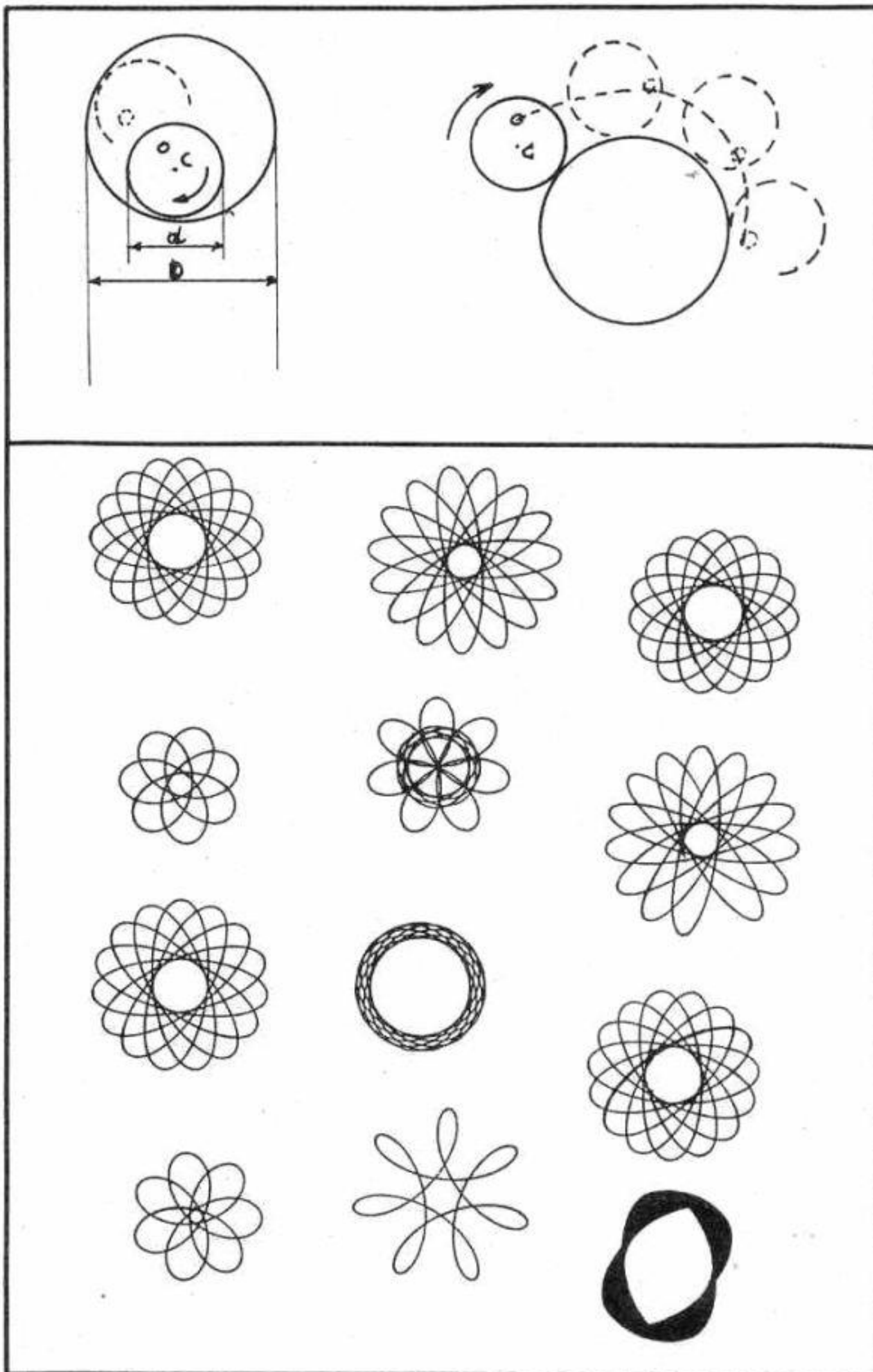
- Per ciò che riguarda il C64 verranno presi in considerazione listati che utilizzano le nostre routine grafiche pubblicate sull'ormai esaurito N. 14 e (udite, udite) anche in Simon's Basic. Listati che utilizzano altre utility grafiche non potranno esser presi in considerazione.

Da che parte cominciare?

Diamo solo un piccolo suggerimento. Le curve ad evolvente sono studiate in modo approfondito nella progettazione delle ruote dentate. Correte dunque in biblioteca e cercate le formule matematiche da applicare, sui testi di disegno meccanico o di costruzioni di macchine.

Le curve di Peano

Ricorsività e merletti in Basic

Ecco come è andata a finire la "sfida" lanciata da C.C.C. nel numero 27.

Bene, anzi molto bene: un sincero grazie da parte della Redazione per la massiccia partecipazione a questo concorso "non concorso" da parte dei lettori.

Sono infatti giunte in Redazione centinaia di lettere o cartoline e addirittura telegrammi (!) con la soluzione.

La maggior parte dei principianti ha indovinato inviando le risposte corrette:

Tipo di trasformazione: 3
Numero di curve: 3
Numero d'ordine: 4

Come avevano promesso, avremmo dovuto inviare un pacco contenente libri, cassette ed altro, soltanto al PRIMO che avesse spedito la soluzione esatta, ma ci siamo resi conto che molti lettori sarebbero stati svantaggiati nei confronti di altri. Mentre, infatti, la consegna della nostra rivista nelle edicole di Milano è quasi immediata, non si può dire lo stesso per altre città e regioni in generale (vedi Sardegna e Sicilia).

Le prime risposte esatte sono arrivate, come prevedibile, da Milano e di conseguenza ci siamo posti il problema:

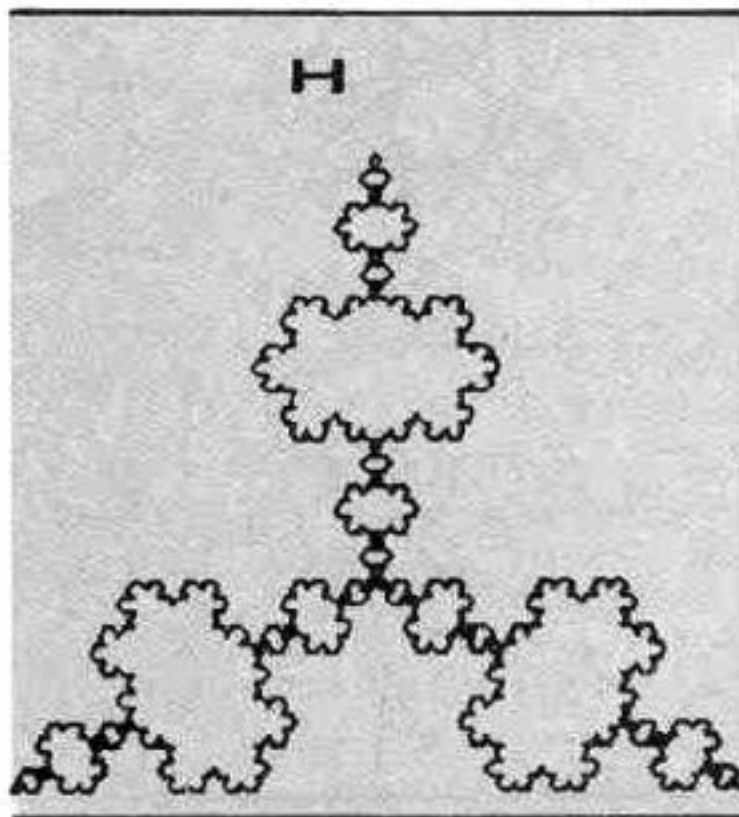
E' giusto premiare il PRIMO (sicuramente un milanese) o non sarebbe meglio operare una specie di "sorteggio" ed estrarre i nomi di più lettori ai quali inviare premi?

Abbiamo votato per la seconda soluzione scegliendo, tra le risposte prevenute, quelle di quaranta lettori (40 invece di uno solo) e questo anche per manifestare il nostro piacere nel vedere che simili iniziative incontrano successo.

E' stata operata una selezione preliminare fra tutte le soluzioni, non solo per stabilire le risposte esatte ma anche per premiare i più meritevoli.

E chi sono i più "meritevoli"

Bhe, innanzi tutto coloro che abitano nel Sud (e sapendo quindi di ricevere la rivista con ritardo rispetto ai lettori del Nord) hanno deciso di concorrere ugualmente pur consci del loro considerevole svantaggio.

In secondo luogo coloro che non possedendo un C–64 ma un C–16 o Plus 4 hanno convertito le istruzioni originali per addattarle al Basic 3.5.

In terzo luogo l'unica ragazza di cui abbiamo ricevuto la soluzione (chissà perchè le ragazze che ci leggono sono numericamente inferiori ai ragazzi: l'informatica non è mica per soli uomini...).

Un premio a parte è stato inviato ad una persona che, a nostro parere, è la più meritevole in assoluto.

Stiamo parlando di un detenuto che ci legge puntualmente e, pur non avendo la possibilità di sperimentare effettivamente su computer la soluzione (nel carcere in cui è rinchiuso è vietato l'uso di personal computer) ha letto attentamente l'articolo, ha studiato il programma, ha "simulato" con carta e penna la soluzione (! !) e ha inviato la lettera contenente la risposta.

Purtroppo uno dei parametri era sbagliato, ma nonostante ciò abbiamo deciso di premiarlo ugualmente (con un abbonamento annuale) per gratificarlo del suo impegno.

In conclusione, non possiamo che ringraziare nuovamente i nostri lettori promettendo in futuro di promuovere altre iniziative simili.

La Redazione

Lire **10.000**

Commodore COMPUTER CLUB

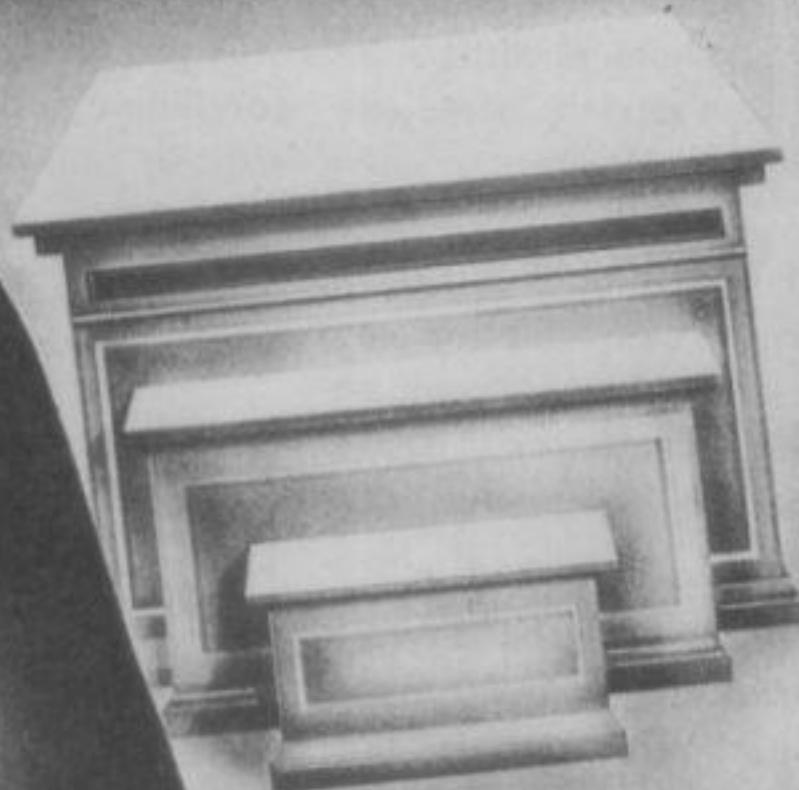
presenta

Commodore Club SPECIAL

Suppl. al n 4

Fantastico!

Disegna col solo joystick sullo schermo del tuo 64

RAFFAELLO 64

Eliana: il LOGO di Commodore Computer Club

Un interprete, a basso costo, ideale per avvicinare i bambini al mondo dell'informatica

Sul numero scorso, dedicato in gran parte ai linguaggi, abbiamo parlato di un particolare linguaggio, il LOGO appunto, che ha suscitato l'interesse di numerosi lettori curiosi di sapere se è proprio vero che, con programmi di tal genere, è possibile introdurre perfino un bambino nel mondo dei computer.

Di programmi (o meglio, linguaggi) di semplice apprendimento, ne sono stati commercializzati parecchi, idonei a girare su molti computer. Quasi tutti, purtroppo, accettano solo comandi in inglese e, se non bastasse, sono spesso venduti con istruzioni nella medesima lingua. Se aggiungiamo che il prezzo non risulta del tutto contenuto, ecco spiegato perché un linguaggio ideale per i superprincipianti non ha incontrato il successo che ci si aspettava.

Esattamente un anno fa la nostra casa editrice propose un linguaggio molto valido per chi avesse avuto intenzione di accostarsi al mondo dell'informatica in modo semplice ed efficace. Danilo Toma, uno dei nostri più validi collaboratori, mise insieme un set di istruzioni grafiche in alta risoluzione, una struttura logica a prova di errori, un gruppo di comandi ed istruzioni in italiano ed implementò il tutto in un programma linguaggio cui diede il nome di E.LI.ANA. (Elementare Linguaggio ANAlogico).

Questo programma, che gira solo sul Commodore 64, fu commercializzato su nastro. Sulla stessa cassetta trovarono posto, inoltre, altri cinque programmi per C64 (Cover, Flipper, Fantatron, Slot Machine), quattro programmi per Vic20 (Cover, Cobra, Circuito, Spaccamuri) e tre per C16 e Plus/4 (Cover, Labirinto, Poker).

Il prezzo del prodotto (solo L. 6500) era più che accessibile al grosso pubblico.

La confenzione conteneva, ovviamente, un elenco delle istruzioni che consentivano di spostare la tartarughina sullo schermo del computer in base ad una semplice sintassi di immediata comprensione.

Il "discorso" sul linguaggio destinato ai piccoli utenti (e, come abbiamo avuto modo di constatare, anche agli adulti!), non si esaurì con la pubblicazione della cassetta.

Sollecitati, infatti, dai lettori, per illustrare in modo più ampio le potenzialità del linguaggio proposto, destinammo, su Commodore Computer Club, una vera e propria trattazione a puntate che trovò spazio sul N. 24 ("Imparando ad imparare"), sul N. 25 ("Il paese elettronico"), sul N. 26 ("Ed il video fa quaderno") e sul N. 27 ("Una stellina passo dopo passo").

I comandi di E.LI.ANA.

Le istruzioni che costituiscono il set del linguaggio sono idonee, principalmente, per spostare sullo schermo una piccola coccinella che lascia, nel corso del movimento, una scia dietro di sè. Tale accorgimento, di render disponibili istruzioni di carattere prevalentemente grafico, riesce più di ogni altro a rendere comprensibile il senso dell'istruzione stessa. Un bambino, infatti, non solo è in grado di valutare la corretta esecuzione di un ordine ma, soprattutto, si diverte nell'osservare lo spostamento ordinato che, quasi sempre, genera percorsi molto belli a vedersi.

LISTA. Consente di esaminare l'elenco delle istruzioni che la coccinella deve compiere.

ESEGUI. La coccinella esegue gli ordini memorizzati.

BASIC. Si abbandona E.LI.ANA. per ritornare al Basic.

SINISTRA e analogamente: DESTRA, AVANTI, INDIETRO, VELOCITA'. Consentono di impostare la direzione e la velocità del movimento da compiere.

MATITA. La coccinella, camminando, lascia la scia dietro di sè.

VOLA. Come è intuitivo, questo è il comando per spostare la coccinella da un punto all'altro dello schermo senza lasciar traccia.

GOMMA, CESTINO. Sono istruzioni valide per cancellare, in tutto o in parte, il disegno effettuato.

ALBUM, QUADERNO. Permettono la gestione del "foglio" da disegno o della lista degli ordini da eseguire.

Molte altre istruzioni consentono di digitare linee di comandi, di calcolare, di assegnare variabili, di cancellare parti di programma, di utilizzare sottoprogrammi.

Sono ovviamente presenti istruzioni specifiche per la memorizzazione e il caricamento su supporto magnetico.

Come procurarsi il Logo

I lettori interessati ad entrare in possesso della cassetta in oggetto, devono inviare la modica cifra di L. 8000 comprensive di spese di spedizione. Non ci è possibile, infatti, inviare materiale in contrassegno.

Compilate un normale modulo di C/C postale indirizzando a:

C/C postale N. 37952207
SYSTEMS Editoriale
Viale Famagosta, 75
20142 Milano

Non dimenticate di indicare chiaramente, sul retro del modulo (nello spazio indicato con "Causale del versamento") non solo il vostro nominativo completo, ma anche la cassetta desiderata: Cassetta "Commodore Club" N. 6.

C 16 - PLUS 4

Vai col Boogie - Woogie!

C.C.C. ha spesso trattato argomenti riguardanti la musica. In questo articolo si affronta uno dei temi più richiesti dai lettori: le potenzialità sonore del C-16 e del Plus/4

di Diego e Federico Canetta

Vi siete mai chiesti che cosa differenzia un suono da un rumore? Noi sappiamo benissimo distinguere un brano musicale dal rombo di un motore di automobile, il cinguettio di un passero dalla scarica di una mitragliatrice, siamo insomma in grado di decidere se una vibrazione sonora è musica oppure rumore. Spesso però non sappiamo dare una spiegazione del fenomeno, che peraltro è molto semplice: nel caso di brani musicali tutte le frequenze che compongono i suoni sono frequenze ben precise ed in relazione fra loro, mentre, nei rumori, molte frequenze casuali diverse si sovrappongono tra loro generando suoni che appaiono "sgradevoli".

Nel C-16 è possibile generare suoni, con frequenze ben definite, e rumori, cioè frequenze tali da riprodurre quello che viene detto "rumore bianco".

I generatori sonori del C-16

Il C-16 possiede notevoli capacità sonore che gli vengono conferite da due generatori, di cui il primo produce solo suoni mentre il secondo può produrre sia suoni che rumori.

E' così possibile riprodurre

brani musicali ad una o due voci, oppure simulare, ad esempio, la caduta della pioggia od il "bang" di un aereo.

Utilizzare i generatori sonori è molto semplice poichè, a differenza di altri computer, la loro gestione può essere effettuata via Basic. Le istruzioni da usare sono essenzialmente due:

• *L'istruzione VOL, seguita da un numero compreso tra 0 ed 8, che permette di selezionare il volume desiderato senza esser costretti, tra l'altro, a regolare manualmente la manopola del TV.*

• *L'istruzione SOUND, seguita da tre valori, che segnalano al computer rispettivamente il generatore a cui ci si riferisce, la frequenza della nota e la sua durata.*

Nell'istruzione VOL il valore zero corrisponde al volume nullo per cui non sarà emesso alcun suono, mentre il valore otto corrisponde al volume massimo.

Nell'istruzione SOUND il primo numero si riferisce al generatore scelto e può valere 1 (generatore numero 1), 2 (generatore numero 2 utilizzato come generatore di suoni) oppure 3 (generatore numero 2 utilizzato come generatore di rumore).

Il secondo numero si riferisce alla frequenza della nota generata e può variare da 0 a 1015. Nella tabella pubblicata è riportata la corrispondenza tra il numero fornito e la frequenza riprodotta.

Il terzo numero riguarda la durata della nota e può variare 0 e 65535. L'unità di misura del tempo per il C–16 è il sessantesimo di secondo. Attenzione a non porre numeri troppo bassi o troppo alti: se ponete 0 la nota non verrà riprodotta, mentre se ponete 65535, durerà ben 16 minuti!

Un semplice programma musicale

La possibilità di controllare i generatori di suoni via Basic permette di costruire programmi in grado di riprodurre simpatiche musichette.

Per dimostrarvi come sia facile comporre un brano musicale con il C16, abbiamo preparato un breve programma che suona un boogie-woogie. Nel programma sono inserite alcune opzioni che permettono di variare la velocità di esecuzione della musica e l'inserimento della seconda voce insieme alla prima, il tutto senza fermare l'esecuzione.

ALLARME ROSSO

(per i principianti)

Nel digitare righe di programma basic che contengono istruzioni DATA, è piuttosto facile incorrere in errori di digitazione. Supponiamo che un'ipotetica linea basic numerata con 1200 debba contenere i tre valori: 123, 456, 789. Ecco alcuni esempi di errori più frequentemente commessi

```
1200 DATA,123,456,789
```

C'è una virgola dopo la parola "DATA". I dati letti dal computer sono, in questo caso, quattro: 0, 123, 456, 789. Se, infatti, non figura alcun carattere dopo l'istruzione DATA, automaticamente viene assunto il valore nullo (0)

```
1200 DATA 123,456,789,
```

In questo caso, dopo il numero 789, il computer, grazie alla presenza della virgola erroneamente inserita, "crede" che ci sia un altro valore e, non trovandolo, lo assume come nullo (0)

```
1200 DATA 1234,56,789
```

La virgola è posizionata male, vale a dire dopo il carattere "4" e non dopo il carattere "3". Il computer non può sapere se il valore esatto è 123 oppure 1234 e individuare un errore, in questo caso, risulta piuttosto laborioso

Come funziona il programma

Riga 10: il volume viene posto a 8 e si assegnano convenienti valori alle variabili "SI" (SI=1 seconda voce disinserita, SI=0: seconda voce inserita) e B (durata della nota).

Righe 20–80: stampa sullo schermo delle opzioni possibili.

Riga 90: lettura dei DATA; se A=0 il controllo del programma passa alla riga 160 che resetta i DATA all'inizio della riga 170.

Riga 100: riproduzione dei suoni. Si noti che la seconda voce è posta ad una frequenza leggermente più bassa della prima.

Righe 110–130: vengono interpretate le eventuali opzioni.

Infine...

Come avete potuto vedere; la gestione delle musiche con il C–16 è estremamente semplice. Con poche istruzioni si riproducono brani musicali più che dignitosi.

Nota	Valore	Freq. (hZ)	Nota	Valore	Freq. (hZ)	Nota	Valore	Freq. (hZ)
la	7	110	re	643	293,6	sol	881	782,2
si	118	123,5	mi	685	330	la	897	880,7
do	169	130,8	fa	704	349,6	si	911	989,9
re	262	146,8	sol	739	392,5	do	917	1045
mi	345	164,7	la	770	440,4	re	929	1177
fa	383	174,5	si	798	494,9	mi	939	1316
sol	453	195,9	do	810	522,7	fa	944	1398
la	516	220,2	re	834	588,7	sol	953	1575
si	571	246,9	mi	854	658			
do	596	261,4	fa	864	699			

Secondo registro musicale C-16 Plus/4

```
1 REM BOOGIE - WOOGIE
2 REM MUSICA PER C-16
3 REM E PLUS-4
4 :
5 REM DIEGO E FEDERICO CANETTA
6 :
10 VOL8:SI=1:B=10:C$=CHR$(13)
20 PRINTCHR$(147)"RITARDO:"
25 PRINTSPC(6)" ┌─────────────────────────┐"
26 REM SHIFT + "U" + "C" + "I"
30 PRINTSPC(6)"|        OPZIONI          |"
31 REM SHIFT + "B"
40 PRINTSPC(6)"|A=AUMENTA                |"
50 PRINTSPC(6)"|D=DIMINUISCE             |"
60 PRINTSPC(6)"|S=INSERISCE VOCE 2       |"
70 PRINTSPC(6)"|N=DISINSERISCE VOCE 2    |"
80 PRINTSPC(6)" └─────────────────────────┘"
81 REM SHIFT +"J" + "C" + "K"
90 READA:IFA=0THEN160
100 PRINTCHR$(19)SPC(8)INT(B*100)/100;
101 PRINTCHR$(157)"--NOTA:"ACHR$(157)"--"
105 SOUND1,A,B:IFSI=0THENSOUND2,A-10,B
110 GETA$:IFA$="S"THENSI=0
120 IFA$="N"THENSI=1
130 IFA$="A"THENIFB>0THENB=B*.85
140 IFA$="D"THENB=B+1
150 GOTO90
160 RESTORE170:GOTO90
170 DATA169,345,453,516,540,516,453,345,169
175 DATA345,453,516,540,516,453,345,383,516
180 DATA596,643,665,643,596,516,169,345,453
185 DATA516,550,516,453,345,453,571,643,571
190 DATA383.516,596.516,0
200 END
```

Tutti i caratteri del Commodore 64

Come ridefinire i caratteri del popolare computer, come colorarli fino a quattro colori, come spostarli all'interno della memoria. Un trattato completo, insomma, per torturare sadicamente, a fin di bene, il vostro calcolatore...

di Pasquale D'Andreti

Nel primo articolo della serie dedicata alla grafica del 64 (C.C.C. N.28), parlammo di come è organizzata la memoria gestita dal circuito integrato 6567 e come programmarne i registri per attivare le funzioni normalmente irraggiungibili con il solo Basic 2.0.

Le opzioni speciali, di cui descrivemmo il funzionamento e fornimmo alcuni esempi pratici, riguardavano la gestione della grafica in alta risoluzione e il diverso significato assunto dalla memoria di schermo a seconda ci si trovasse in Modo Testo o in Modo Alta Risoluzione.

Come necessario completamento, fornimmo alcune tabelle per la gestione degli spostamenti della memoria adibita alla rappresentazione grafica dei dati con le relative spiegazioni d'uso.

Per coloro che non posseggono il numero in questione, riepilogheremo brevemente , nel riquadro di queste pagine, la parte dedicata

allo spostamento della memoria di schermo tralasciando il resto in quanto non risulta critico per l'apprendimento di quanto fra poco esporremo.

Ciononostante, consigliamo vivamente, a chi può, di dare una ripassata più... sostanziosa alla prima parte.

Maschera	Byte 53271	Inizio schermo	Posiz.
0	00000101	0	0
16	00010101	1024	1
32	00100101	2048	2
48	00110101	3072	3
64	01000101	4096	4
80	01010101	5120	5
96	01100101	6144	6
112	01110101	7168	7
128	10000101	8192	8
144	10010101	9216	9
160	10100101	10240	10
176	10110101	11264	11
192	11000101	12288	12
208	11010101	13312	13
224	11100101	14336	14
240	11110101	15360	15

Tabella 1

Tutti i colori del buio: il Multicolor del Modo Testo

I caratteri del Commodore 64 sono definiti da una sequenza di byte divisi in gruppi di otto; questi byte sono stati registrati, in fabbrica, su di una ROM interna al computer e occupano, in totale, 4 Kbyte.

Il carattere che rappresenta la lettera "A", ad esempio, è situato negli otto byte che vanno da 53256 a 53263; il valore binario di questi ultimi è riportato nella figura 1.

Nel modo testo normale, ogniqualvolta il Vic II deve visualizzare la lettera "A" maiuscola, preleva questi otto byte, li legge bit per bit e disegna un puntino (dot) con il colore dello sfondo se il bit appena analizzato vale 0, mentre, se il bit vale uno, il puntino sarà del colore il cui codice è situato nella relativa casella del colore.

Nel modo testo multicolor, invece, anzichè leggere i bit uno per volta, il Vic II li legge in coppia, cosicchè un carattere non sarà composto da 8 puntini orizzontalmente e 8 verticalmente, ma i puntini che lo formeranno in orizzontale saranno 4, ciascuno, però, di larghezza doppia.

Anche il modo di assegnare i colori cambia e, a seconda del valore dei due bit letti (vedi tabella 2), il C—64 disegna un puntino (largo, lo ripetiamo, come due puntini "normali") il cui colore può essere prelevato da una delle quattro "fonti" di colori disponibili:

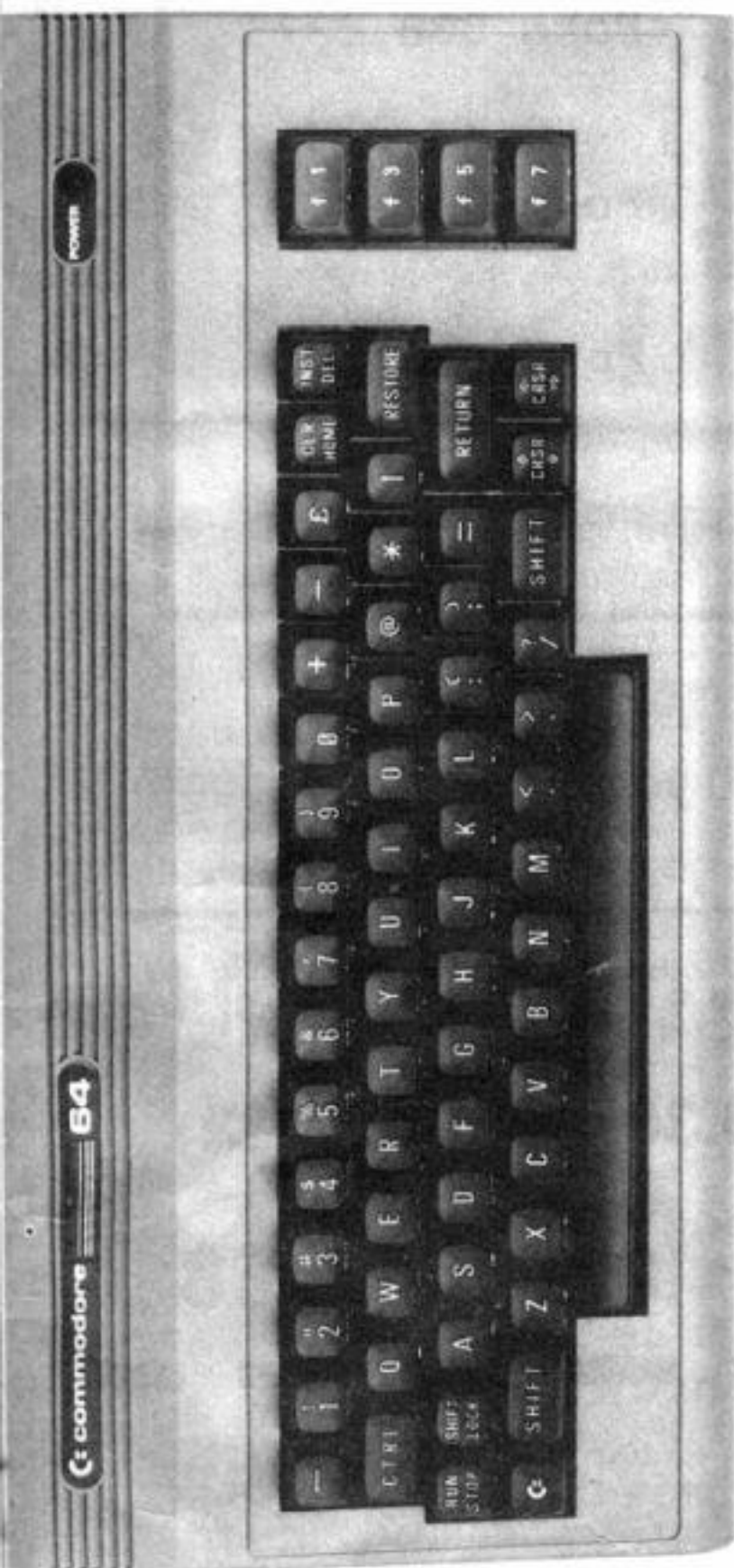
Coppia di bit	Col. n.	Indir. colore
00	1	53281
01	2	53282
10	3	53283
11	4	Cella colore (55296-56295)

Tabella 2

- *la prima è la locazione che contiene il colore di sfondo dello schermo.*
- *la seconda è la casella della normale memoria del colore.*
- *la terza e la quarta fonte sono rappresentate da altri due byte, non usati nel modo "normale", riportati nella tabella 2.*

Considerando che il codice del colore di sfondo risiede nella locazione 53281 e gli altri

```
100 REM  LISTATO 1
110 PRINTCHR$(147)
120 POKE 53270,PEEK(53270) OR 1
    6:REM  ATTIVA MULTICOLOR
130 POKE 2023,1:REM  PONE CARAT
    TERE "A" IN BASSO A DESTRA
140 POKE 53281,0:REM  ASSEGNA C
    OLORE NERO ALLE COPPIE 00
150 POKE 53282,11:REM  ASSEGNA
    COLORE GRIGIO1 ALLE COPPIE
    01
160 POKE 53283,12:REM  ASSEGNA
    COLORE GRIGIO3 ALLE COPPIE
    10
170 POKE 56295,15:REM  ASSEGNA
    COLORE GRIGIO3 ALLE COPPIE
    11
180 FOR I=0 TO 255:POKE 1024+I,
    I:POKE 55296+I,15:NEXT
190 GET A$:IF A$="" THEN 190
200 POKE 53270,PEEK(53270) AND
    239:REM  DISATTIVA MULTICOL
    OR
```

COME METTERE D'ACCORDO UN DO CON UN BIT.

No, non è fantascienza. Stiamo solo parlando delle fantastiche tastiere elettroniche Antonelli. Una novità che ti permette di dialogare con l'interfaccia di un computer per comporre, memorizzare e riascoltare tutta la musica che vuoi. E questo è solo l'inizio. Le tastiere elettroniche Antonelli sono una vera orchestra con tantissimi strumenti e voci diverse, possono essere collegate a impianti hi-fi per ottenere una fedeltà e una potenza di suono mai viste. In più sono facili da usare e belle da mostrare agli amici. Il prezzo? Questo sì è da fantascienza: da L. 250.000 (IVA esclusa).

Vuoi sapere di più sulle tastiere elettroniche Antonelli?

Compila e spedisci questo coupon a: ANTONELLI
Via Filottrano - 60027 Osimo (AN). Riceverai un completo catalogo.

NOME ____________ COGNOME ____________

VIA ____________________________ N° ______

CAP. ________ CITTA' ____________________

Antonelli 2614 STEREO

Locaz.	Valore binario	Carattere visualizz.
53256	00011000	**
53257	00111100	****
53258	01100110	** **
53259	01111110	******
53260	01100110	** **
53261	01100110	** **
53262	01100110	** **
53263	00000000	

Figura 1

Codice caratteri	Gruppo	Colore sfondo
0- 63	1	53281
64-127	2	53282
128-191	3	53283
192-255	4	53284

Figura 2

due byte interessati sono il 53282 ed il 53283, i 4 colori sono individuati completamente dalla tabella 2.

Per attivare il modo multicolore bisogna settare (=porre a 1) il bit 4 della locazione 53270 per mezzo dell'istruzione:

POKE53270,PEEK(53270)OR16

Per disattivare il multicolor, basterà digitare:

POKE53270,PEEK(53270)AND239

Se volessimo, ad esempio, far comparire una "A" in basso a destra dello schermo in modo multicolor e assegnare alle coppie di bit 00, 01, 10 e 11 rispettivamente i colori nero, grigio1, grigio2 e grigio3, dovremmo digitare il programmino N.1.

A che cosa può servire l'insolita tecnica di programmazione appena descritta?

In effetti sono piuttosto rari i casi in cui, limitandosi a gestire "solo" il multicolor(senza affiancare altre tecniche), si riescono a creare effetti nuovi o utili.

Fondamentale, però, risulta la gestione del multicolor nella ridefinizione dei caratteri. In questo caso, infatti, si riescono a creare disegni piuttosto elaborati da un punto di vista cromatico.

Il Modo Extended Background

Il modo Extended Background (d'ora in poi indicato semplicemente con: E.B.) permette di dotare ogni singolo carattere di uno sfondo di colore diverso (scelto fra 4) pur mantenendo la quota di 8 dot orizzontali per ciascun carattere.

Il prezzo da pagare, questa volta, consiste nella riduzione dei caratteri rappresentabili, che, da 256, scendono a 64.

In pratica, con l'attivazione del modo E.B, i caratteri si dividono in 4 gruppi di 64 caratteri ognuno, i quali sono identici tranne che per il colore dello sfondo. I caratteri rappresentati sono quelli che vanno dalla "A" commerciale (chiocciolina @) al punto interrogativo (?). La definizione del colore del carattere rimane affidata alla memoria del colore residente da 55296 a 56295. In definitiva si può scrivere in ogni cella dello schermo un qualsiasi carattere, scelto tra 64, colorato con un colore, scelto tra 16, posizionato sopra uno sfondo, scelto tra 4.

I codici dei 4 colori sono quelli rappresentati dal contenuto delle locazioni numerate da 53281 a 53284 secondo lo schema di figura 2.

Per rendere operativo il modo E.B. è necessario porre a 1 il bit 6 della locazione 53265. Per fare questo useremo:

POKE53265,PEEK(53265)OR64

Viceversa, per disattivare l'E.B. metteremo a 0 lo stesso bit della stessa locazione:

POKE 53265, PEEK (53265) AND 191

Apprese le caratteristiche del modo E.B.

```
100 REM  LISTATO 2
110 POKE 53265,PEEK(53265) OR 6
    4:REM  ATTIVA MODO E.B.
120 POKE 53281,1:REM  PONE IL B
    IANCO COME SFONDO 1
130 POKE 53282,15:REM  PONE IL
    GRIGIO3 COME SFONDO 2
140 POKE 53283,12:REM  PONE IL
    GRIGIO2 COME SFONDO 3
150 POKE 53284,11:REM  PONE IL
    GRIGIO1 COME SFONDO 4
160 FOR I=0 TO 15:FOR K=0 TO 25
    5
170 POKE 1024+K,K:POKE 55296+K,
    I:NEXT
180 FOR T=1 TO 3000:NEXTT,I
190 POKE 53265,PEEK(53265) AND
    191:REM  DISATTIVA MODO E.B
```

verifichiamo, digitando il semplice programma N.2, quanto esposto finora.

La ridefinizione dei caratteri

Come abbiamo già detto, il Vic II preleva l'"immagine" dei caratteri da mostrare sul video nella memoria ROM (situata a partire da 53248), a gruppi di otto byte per ciascun carattere.

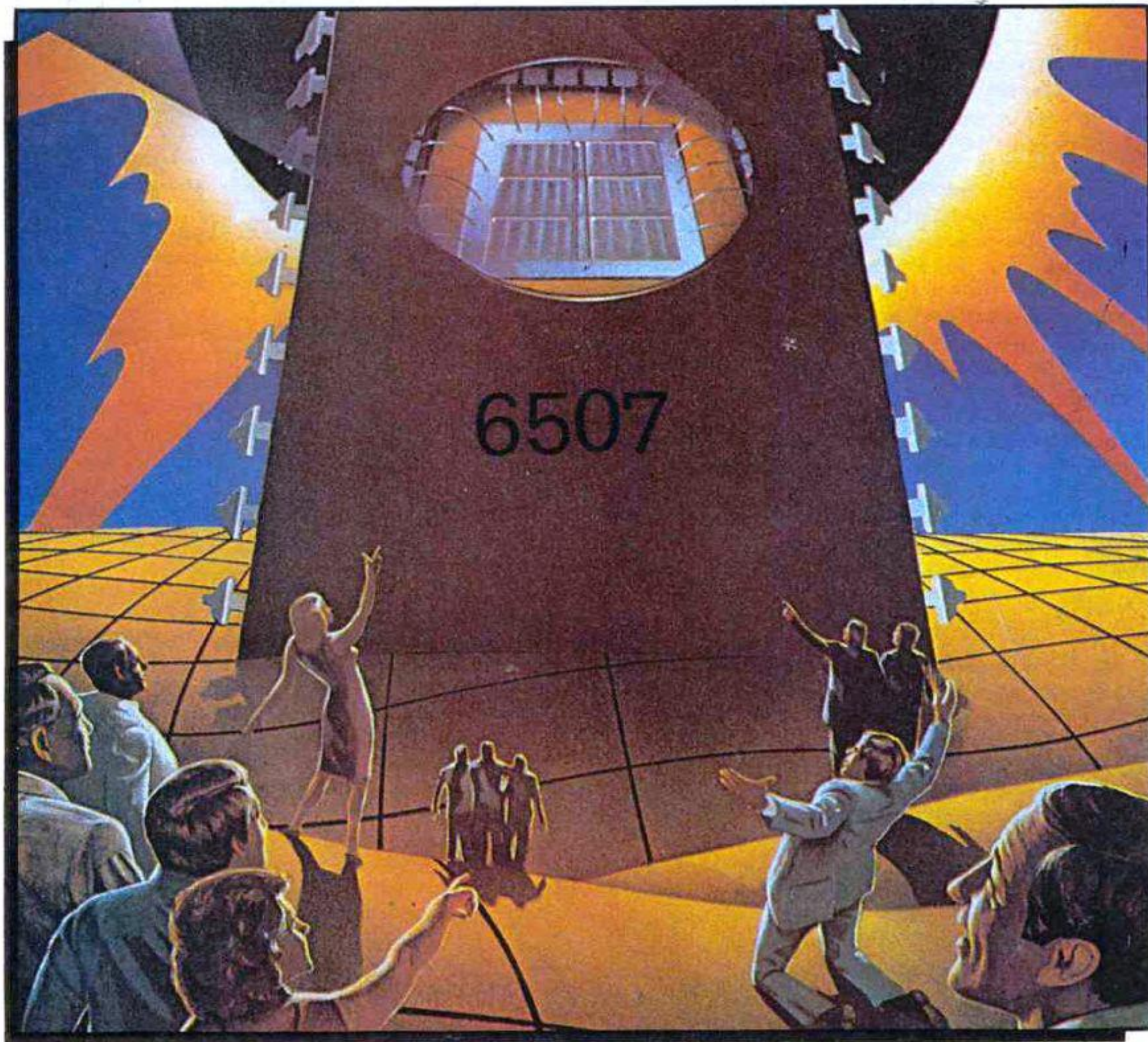
Il computer può visualizzare due set di 256 caratteri cadauno (128 "normali" e 128 in reverse). Dato che ogni carattere è rappresentato da 8 byte, si deduce che sono necessari 4K byte di memoria (256x2x8=4096 cioè 4K) per contenere le immagini binarie dei simboli che appaiono sul nostro video.

Per visualizzare grafici, simboli speciali, disegni e tutto ciò che non sia rappresentabile con i normali caratteri alfanumerici e grafici, si ricorre, in genere, alla grafica in alta risoluzione che, all'innegabile vantaggio di una definizione molto particolareggiata dell'oggetto da descrivere, unisce però una gestione difficile e lenta; senza contare l'estrema difficoltà di scrivere frasi costruite con i caratteri normali insieme ai disegni.

Quando l'oggetto da rappresentare non ha bisogno di molti dot (puntini elementari) e ha buone probabilità di essere ripetuto sullo schermo nel corso del programma, in posizioni o tempi differenti, allora l'operazione da effettuare è, senz'altro, la ridefinizione del set di caratteri.

Ciò è reso possibile da un particolare registro del 6567 che comunica all'integrato la posizione esatta dei 2K che devono essere considerati come "attuale" mappa dei caratteri.

In altre parole, dato che un carattere non è altro che un gruppo di otto byte, sarà possibile, grazie ad istruzioni POKE, "creare" nuovi caratteri in una zona RAM lunga 2K e, subito dopo, comunicare al calcolatore che deve considerare la nostra zona RAM come riferimento per fare apparire i caratteri.

Modificando opportunamente il registro

La memoria di schermo

Lo schermo che all'accensione del computer ci viene mostrato, è formato da 40 colonne e 25 righe, per un totale di (40x25) 1000 caratteri. Inoltre ogni carattere può avere un differente colore a scelta fra 16: in questo modo si può intuire che la memoria occorrente per la visualizzazione di 1000 caratteri, ognuno con un proprio colore, sia di 2000 byte.

Il circuito integrato Vic II (6567) può gestire "solo" 16384 byte per volta (cioè 16K, valore dato da: 2 elevato alla 14ma potenza) e, nel modo di funzionamento "normale", la memoria interessata è quella che va da 0 a 16383. Questo è il primo dei quattro blocchi da 16K che il Commodore 64 possiede (16*4=64K).

Questo banco di memoria, che si può cambiare agendo opportunamente sulla locazione 56576, non viene, in genere, utilizzato per intero e, infatti, la memorizzazione di un programma Basic incomincia dalla locazione 2048 senza interferire con il regolare funzionamento del Vic II.

All'accensione, i 1000 byte occorrenti alla memoria di schermo vengono posizionati a partire da 1024 (la posizione può essere cambiata, sempre nell'ambito dei 16K scelti, agendo sulla locazione 53272) mentre quelli dedicati alla memoria del colore risiedono permanentemente nelle 1000 locazioni che vanno da 55296 a 56295 e non possono essere modificati.

Spostando la memoria di schermo si deve prestare attenzione a non far collidere la memoria dedicata al Basic con quella dedicata allo schermo.

A questo scopo è sufficiente modificare i puntatori 43—44 e 55—56 che indicano, rispettivamente, l'inizio e la fine dell'area di memoria riservata al programma Basic. E' inoltre necessario adeguare il puntatore locato in 648 che è usato dal sistema operativo per conoscere la locazione iniziale dello schermo.

Apprese le parti salienti di ciò che era stato scritto nell'articolo precedente, ci soffermeremo sul modo Multicolor e sull'Extended Background prima di passare all'argomento più interessante, ossia la descrizione del funzionamento dei caratteri ridefinibili.

6567, in modo da farlo "puntare" alla zona di memoria RAM dove avevamo precedentemente riportato i caratteri personalizzati, il Vic II andrà a leggere queste locazioni e riporterà sul video i nostri caratteri invece di quelli standard.

Questo modus operandi viene generalmente utilizzato quando si ha bisogno di rappresentare simboli matematici speciali, alfabeti particolari, apici, pedici e nuovi caratteri grafici mediante i quali ottenere disegni più complessi.

Esempi di caratteri ridefiniti sono presenti in moltissimi programmi commerciali: si va dai giochi, dove sono usati prevalentemente per i fondali in cui si svolge l'azione, ad alcuni word processor italiani, che li usano per ridefinire le vocali accentate ed altri simboli, da quelli di grafica finanziaria a quelli di utilità.

La struttura del nuovo set di caratteri dovrà rispettare la regola secondo la quale ogni carattere ha bisogno di otto byte consecutivi il cui indirizzo iniziale è dato da:

inizio=(inizio del nuovo set) + ((codice carattere) * 8)

Se, ad esempio, il nostro set incomincia a partire da 51200 e il simbolo cercato è stato sostituito alla lettera "P", il cui codice di schermo equivale a 16 (vedere l'Appendice E del manuale del C—64), troveremo la sua rappresentazione nei byte che vanno da 51328 (dato da 51200+(16*8)) a 51335.

In genere, non serve ridefinire tutti i 256 caratteri di un set, ma è più conveniente trasferire il set normale dalla ROM alla zona RAM scelta, sostituire quelli prestabiliti e modificare il puntatore presente in 53272.

La zona RAM utilizzata dal Vic II, al momento dell'accensione, è quella che va da 0 a 16383; in questa area risiede, tra l'altro, anche la memoria video (1024—2023).

Se si volesse creare un set di caratteri nuovi, bisognerebbe posizionarlo in queste locazioni, col rischio di entrare in un disastroso conflitto con l'interprete Basic che, nemmeno a farlo apposta, utilizza, per l'immagazzinamento dei programmi, proprio gli indirizzi che vanno da 2048 a 40959.

Le soluzioni possibili per questo inevitabile problema sono due:

- *Si riduce convenientemente l'area occupata dal Basic per far posto ai 2K dedicati ai caratteri.*
- *Si sposta il banco da 16K che il Vic II gestisce, a partire da 49152 (anziché da 0), dove risiede una zona RAM di ben 4K inutilizzata.*

La prima soluzione ha due svantaggi: ruba memoria al Basic e ha bisogno di un'attenta ripartizione tra l'area programma e l'area caratteri se non si vuole piombare in uno stato irrisolvibile di blocco del sistema.

L'unico fastidio che comporta la seconda soluzione è quello relativo al diverso posizionamento della memoria video che, da 1024—2023, passa a 50176—51175. Per questo motivo vanno cambiate tutte le POKE che si riferiscono al video.

Riepilogando quanto esposto finora, concludiamo che, per modificare i caratteri del Commodore 64, dovremo compiere, sequenzialmente, le seguenti operazioni:

- *Ricopiare i caratteri dalla ROM alla RAM desiderata.*
- *Modificare quelli voluti.*
- *Spostare la memoria gestita dal Vic II a partire da 49152.*
- *Modificare il puntatore sito in 53272 in modo che indirizzi alla locazione d'inizio del nuovo set di caratteri.*

Dopo aver appreso, teoricamente, il metodo da seguire per una corretta ridefinizione dei caratteri, dovremo risolvere i problemi tecnici che si presentano nel mettere in pratica la procedura descritta nei quattro punti.

La questione della memoria nascosta

La ROM generatrice dei caratteri standard risiede, come già detto, nelle locazioni che vanno da 53248 a 57343.

Nelle stesse locazioni risiedono anche 4K di memoria RAM e i dispositivi di I/O (Input/output) che normalmente sono presenti, ossia lo stesso Vic II, i due CIA (Complex Interface Adapter) e il SID (Sound Interface Device).

Nella composizione grafica dei caratteri, il Vic II, strano ma vero, legge il contenuto della ROM senza interferire con le normali operazioni del microprocessore 6510, il quale "vede" solo la presenza dei chip di I/O, ignorando la ROM e la RAM sottostante.

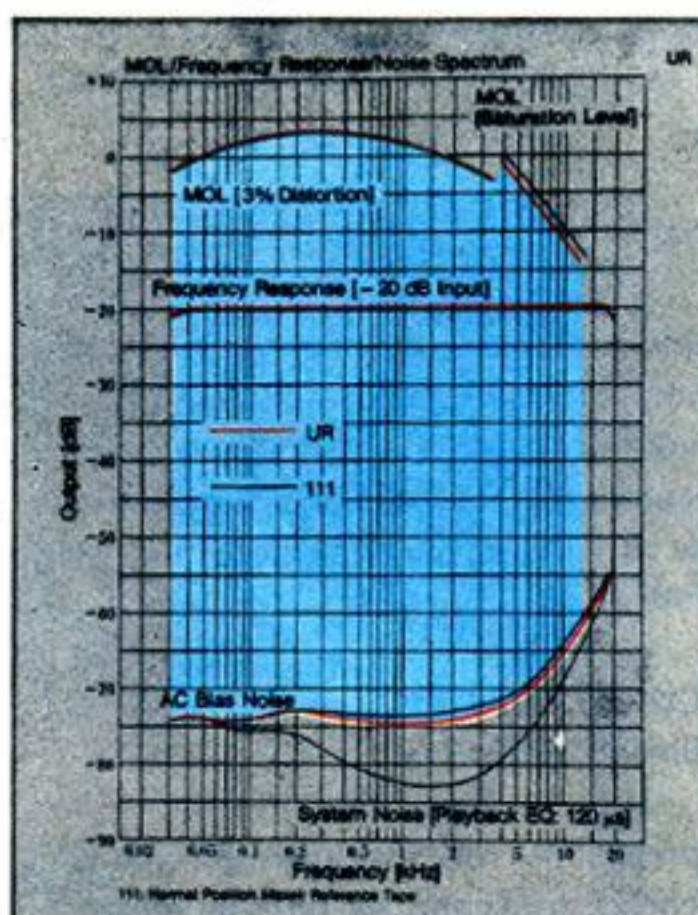
Pertanto, per ricopiare i caratteri standard dalla ROM ad una zona RAM dove poi modificheremo quelli che ci interessano, dovremo "portare alla luce" la ROM, disabilitando l'I/O.

NUOVA MAXELL UR

Una cassetta Low Noise a livello Hi-Fi

C'era una volta la cassetta adatta ad un certo tipo di utilizzo, alla quale non si potevano chiedere prestazioni superiori.

Adesso c'è la UR MAXELL che, grazie alla sua modernissima tecnologia, rivoluziona gli standards della cassetta low noise portandoli a livelli hi-fi. UR significa UNIVERSAL RECORDING, cioè adatta ad ogni tipo di registratore, dal portatile alla piastra ultrasofisticata, sempre con la certezza di prestazioni eccellenti.

Il nastro UR offre un aumento in MOL (Maximum Output Level) di

1,5 dB alle basse frequenze e di 2 dB alle alte rispetto alla serie UL.

L'ulteriore abbassamento del rumore di fondo offre la gamma dinamica più ampia della categoria: fino a 77 dB (a 1 Khz), ottenendo così un suono chiaro e cristallino senza alcuna distorsione.

L'involucro della UR è costruito con un nuovo polistirene di grande resistenza che ne assicura una lunga vita senza problemi.

A voi non resta che provare; non resisterete al fascino, neanche troppo discreto, della nuova UR MAXELL.

maxell®

L'arte di registrare.

DELPI

Così facendo, però, impediremo il flusso regolare degli interrupts provenienti dal timer A del CIA 1 causando il blocco del sistema.

Di questo inevitabile problema (davvero complicato per i principianti) non diamo, per ora, una esemplificazione perchè richiederebbe per lo meno un altro articolo! Ci limitiamo, comunque, a darne la soluzione.

Per fermare temporaneamente il funzionamento del timer A, dovremo mettere a 0 il bit 0 della locazione 56334, mentre per "evidenziare" la ROM dovremo porre a 0 il bit 2 della locazione 1.

Ciò si effettua anche da Basic mediante le poche istruzioni che seguono:

POKE56334,PEEK(56334)AND254
POKE1,PEEK(1)AND251

Per ritornare al modo standard di gestione della memoria, e per ripristinare gli interrupts, dovremo rimettere a 1 i bit prima azzerati in un modo altrettanto semplice:

POKE1,PEEK(1)OR4
POKE56334,PEEK(56334)OR1

Bisogna porre la massima attenzione nel ricopiare queste linee rispettando strettamente la sequenzialità con le quali sono state esposte usandole solo in modo programma. Naturalmente, fra i due gruppi di POKE, metteremo un ciclo For—Next che legga i byte dalla ROM depositandoli in RAM.

Il listato N.3 provvede a compiere l'operazione completa di copiatura del set di caratteri maiuscolo—grafico nella zona RAM che parte da 51200. Per copiare il set minuscolo—maiuscolo, basta sostituire nel PEEK della linea 20 il numero 55296.

Il programma N.3 impiega diverso tempo per effettuare l'operazione di copia, ma ha il vantaggio di essere facilmente modificabile perchè è scritto in Basic. Tuttavia, per motivi di velocità, utilizzeremo, nell'ultimo programma di queste pagine, una brevissima routine in Linguaggio Macchina che farà lo stesso lavoro in circa un decimo di secondo.

Come modificare i caratteri

Una volta ricopiati i caratteri, il discorso diventa molto più semplice ma anche più noioso perchè gran parte del lavoro lo dovremo effettuare noi stessi, armati di carta quadrettata, matita e una dose di pazienza direttamente proporzionale al numero di caratteri che intendiamo riprogrammare.

Innanzitutto dovremo scegliere i caratteri vecchi che dovremo sostituire con i nuovi; in teoria, qualsiasi carattere andrebbe bene, ma per ragioni pratiche, consigliamo di sostituire quei simboli che si usano di meno in condizioni normali, ossia i caratteri in reverse.

Questi ultimi sono, come ricorderete, 128 e i loro codici vanno da 128 a 255; la relazione che lega il carattere normale al suo corrispettivo invertito, è:

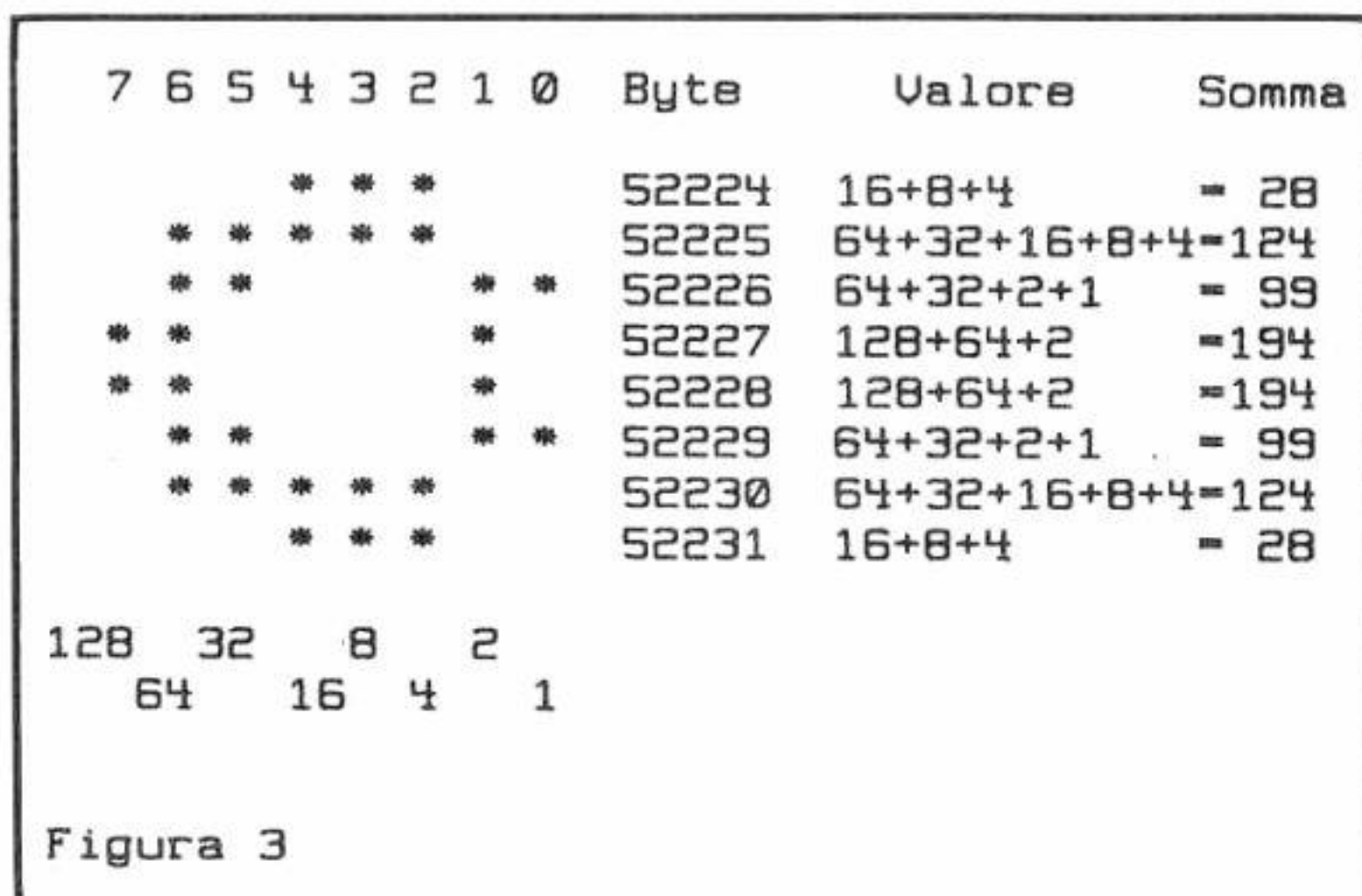
I=N+128

In cui "I" è il codice invertito e N è il codice normale.

Per esempio, siccome il codice di schermo della lettera "C" è 3, il suo invertito sarà 131 (=128+3).

Modificando soltanto i caratteri in reverse (lasciando inalterati quelli normali), avremo il vantaggio di ritenere in memoria i normali caratteri per gli usi di scrittura contemporaneamente a quelli riprogrammati, che si potranno ottenere premendo il tasto Rvs/On

```
100 REM  LISTATO 3
110 PRINTCHR$(142):REM  MAIUSCO
    LO/GRAFICO
120 PRINTCHR$(147) TAB(10)"ATTE
    NDERE 2047"
130 PRINT:PRINT"SE HAI PREMUTO
    RUN/STOP E RESTORE"
140 PRINT:PRINT"BATTI ALLA CIEC
    A: POKE 648,4"
150 REM  DISABILIIA I/O
160 POKE 56334,PEEK(56334) AND
    254:POKE 1,PEEK(1) AND 251
170 REM  TRASFERISCE SET DI CAR
    ATTERI DA 53248-55295  A 51
    200-53247
180 FOR I=0 TO 2047:POKE I+5120
    0,PEEK(I+53248):PRINTCHR$(1
    9)I:NEXT
190 REM  RIPRISTINA I/O
200 POKE 1,PEEK(1) OR 4:POKE 56
    334,PEEK(56334) OR 1
210 REM  CAMBIA PUNTATORE SET C
    ARATTERI
220 POKE 53272,(PEEK(53272) AND
    240) OR 2
230 REM  SELEZIONA BANCO DI MEM
    ORIA
240 POKE 56576,PEEK(56576) AND
    252:POKE 648,196
250 PRINTCHR$(147)"ECCO UNA PIC
    COLA DIMOSTRAZIONE:"
260 PRINT:PRINT"@@@@@@@@@@@@@@@
    @@@":PRINT
270 PRINT"VEDRAI SCOMPARIRE LE
    CHIOCCIOLINE"
280 PRINT"GRAZIE ALL'ISTRUZIONE
     POKE RELATIVA"
290 PRINT"AGLI OTTO BYTE DA 512
    00 A 51207":PRINT
300 PRINT"PREMI UN TASTO"
310 GET A$:IF A$="" THEN 310
320 FOR I=51200 TO 51207:POKE I
    ,0:FOR J=1 TO 1000:NEXTJ,I
330 PRINT"GUARDA ORA:"
340 FOR I=51200 TO 51207:POKE I
    ,255:FOR J=1 TO 500:NEXTJ
350 POKE I,1:FOR J=1 TO 500:NEX
    TJ:NEXTI
```

```
 7 6 5 4 3 2 1 0   Byte    Valore          Somma

       * * *       52224   16+8+4          = 28
   * * * * *       52225   64+32+16+8+4=124
   * *       * *   52226   64+32+2+1       = 99
 * *         *     52227   128+64+2        =194
 * *         *     52228   128+64+2        =194
   * *       * *   52229   64+32+2+1       = 99
   * * * * *       52230   64+32+16+8+4=124
       * * *       52231   16+8+4          = 28

128   32    8    2
   64   16    4    1

Figura 3
```

(Ctrl+9) e il tasto corrispondente al carattere sostituito.

Come abbiamo già detto, per trovare l'inizio degli 8 byte componenti l'immagine del simbolo che vogliamo modificare, basta aggiungere al suo codice moltiplicato per 8, il valore della locazione d'inizio del nuovo set di caratteri.

Una volta individuate le otto locazioni consecutive ove risiede il nostro carattere, dovremo depositarvi gli otto byte che corrispondono alla sua nuova immagine.

Questa operazione viene svolta in modo diverso a seconda si operi con il modo di rappresentazione multicolor, extended background o normale.

Per il modo extended background il discorso fatto prima circa la collocazione dei caratteri deve essere modificato in quanto, come già detto, ne vengono considerati solo i primi 64 e, ovviamente, bisogna lavorare solo su questi.

Per ciò che riguarda il resto, il modo E.B. lavora come quello normale.

Ora prendiamo un foglio quadrettato e definiamo un quadrato di 8 caselle per lato: in questa scacchiera dovremo disegnare il nostro simbolo.

Il modo di procedere per disegnare dei caratteri in modo normale o E.B. si distingue dal modo multicolor per il significato che il Vic II conferisce ai bit accesi e spenti, come abbiamo già accennato nei paragrafi precedenti. Pertanto tratteremo separatamente i casi normale—E.B. da una parte, e multicolor dall'altra, mostrando solo come fare.

Il disegno dei caratteri normali—E.B. avviene nella solita matrice 8x8 nel seguente modo:

- *Si anneriscono le caselle che rappresentano i contorni della figura.*
- *Si assegna, riga per riga, alle caselle annerite il valore unitario (1) e a quelle bianche il valore nullo (0) all'interno del corrispondente byte.*
- *Si trasforma in decimale la stringa binaria ottenuta.*
- *Si depone il valore ottenuto nella locazione corrispondente alla riga (1—8) da cui era stato ricavato.*

Per chiarire le idee, ipotizziamo di voler disegnare, al posto della "A" commerciale in reverse (chiocciolina: @), il cui codice è 128, il logo della Commodore.

Innanzitutto dovremo ricercare l'indirizzo degli otto byte che rappresentano questo simbolo. Se abbiamo trasferito i caratteri a partire da 51200, esso sarà:

51200+(8*128)=52224

Ora che conosciamo la locazione del primo byte tracciamo lo schema di figura 3.

Come è possibile intuire dalla stessa figura, si deve calcolare il valore da POKare nel rispettivo byte mediante la somma dei valori che rappresentano i singoli bit accesi (quelli rappresentati dall'asterisco) in una stessa riga.

Questi valori sono indicati in basso e rappresentano le potenze di 2 i cui esponenti coincidono con la posizione dei bit indicati in alto.

Più a destra sono indicati i byte in cui depositare i valori trovati con questo metodo, corrispondenti a quelli che rappresentavano l'immagine del simbolo della chiocciolina (@) invertito nel nostro nuovo set di caratteri.

Per definire i caratteri multicolor la procedura per calcolare i valori da immettere nei byte è la stessa, ma il Vic II, anzichè leggere un bit alla volta, ne leggerà due (come spiegato nel paragrafo dedicato a questo modo di rappresentazione), visualizzando un dot di larghezza doppia ma con quattro colori rappresentabili.

Naturalmente, la matrice sulla quale disegneremo il nostro carattere, sarà sempre di 8x8 caselle. Tuttavia dovremo considerare ogni coppia di bit orizzontali come un unico puntino. Di conseguenza vedremo apparire un carattere inscritto in una matrice di 8x8 puntini.

Lo stesso carattere di prima, così come è stato disegnato, risulterebbe illeggibile in mo-

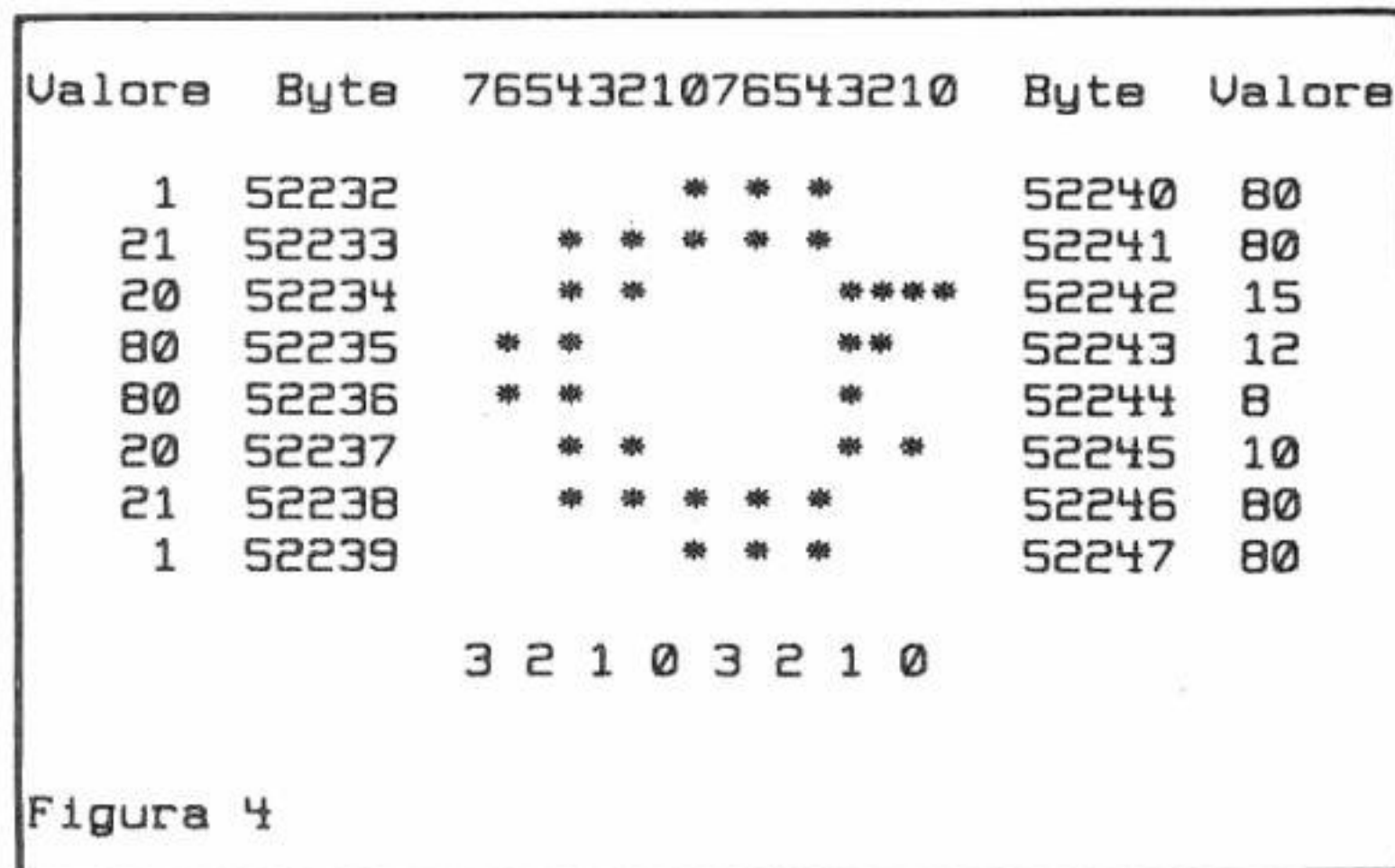
```
Valore  Byte   7654321076543210  Byte   Valore

   1    52232         * * *       52240  80
  21    52233     * * * * *       52241  80
  20    52234     * *       ****  52242  15
  80    52235   * *         **    52243  12
  80    52236   * *         *     52244  8
  20    52237     * *       * *   52245  10
  21    52238     * * * * *       52246  80
   1    52239         * * *       52247  80

               3 2 1 0 3 2 1 0

Figura 4
```

Mappa n.	Bit 1-3	Inizio mappa
0	XXXX000X	0
1	XXXX001X	2048
2	XXXX010X	4096
3	XXXX011X	6144
4	XXXX100X	8192
5	XXXX101X	10240
6	XXXX110X	12288
7	XXXX111X	14336

Tabella 3

do multicolor, in quanto apparirebbe sul video un miscuglio di colori senza senso.

Ma la nostra testardaggine è grande e decidiamo comunque di dotare il nostro 64 del suo logo in multicolor.

Di soluzioni ce ne sono molte, ma la più immediata e semplice nasce dall'idea che, se è vero che un carattere si riduce a 4x8 dot, è anche vero che due caratteri adiacenti formano una matrice di 8x8 dot, cioè proprio quella che ci serve.

Ovviamente, operando in questo modo, allargheremo il carattere del doppio, ma potremo avere a disposizione ben 4 colori visualizzabili contemporaneamente nell'ambito dello stesso carattere.

Nella figura 3 è indicato come dovrebbe apparire il nostro (doppio) carattere, da sostituire al posto della "A" e della "B," in reverse (codici 129 e 130). Si suppone, nella figura indicata, che le coppie 00 rappresentino il bianco, le 01 il nero le 10 il blu e le 11 il rosso.

Ponendo che i due caratteri ai quali sostituire il nuovo simbolo siano "A" e "B" in reverse, dovremo sempre farli apparire adiacenti e nell'ordine giusto per ottenere una visualizzazione leggibile.

Naturalmente, prima di ogni cosa, dovremo attivare il multicolor e scegliere i colori in modo opportuno come indicato dal paragrafo dedicato a questo argomento.

I banchi di memoria

Nell'introduzione abbiamo discusso esaurientemente di come scegliere la zona di memoria destinata a ricevere l'immagine dei nuovi caratteri giungendo alla conclusione che quella più adatta è situata a partire da 49152 a 53247.

Abbiamo anche osservato che la zona RAM utilizzata normalemente dal chip video 6567 è quella che va da 0 a 16347.

Mutando i bit 0 e 1 della locazione 56576 e, in particolare, ponendoli entrambi a zero, si obbliga il Vic II a considerare la zona che va fa 49152 a 65535.

Gran parte di quest'area è occupata da memoria ROM e memoria RAM (contemporaneamente) ma, come vedremo prossimamente, si può fare in modo di creare pagine grafiche anche nella memoria normalmente utilizzata dalla ROM.

Non intendiamo ampliare, per adesso, il discorso dello spostamento dei banchi di memoria video (il problema è stato parzialmente affrontato nell'articolo precedente) ma facciamo osservare che, d'ora in poi, ogni vecchio indirizzo relativo a qualsiasi tipo di memoria grafica (grafica in alta risoluzione, memoria di schermo, puntatori agli sprites ecc.) dovrà essere aumentato della quantità costante 49152.

Per porre a zero i bit 0 e 1 della locazione 49152 basta eseguire la seguente istruzione:

POKE56576,PEEK(56576)AND252

Ma ciò non basta; infatti la routine del sistema operativo che provvede alla scrittura sullo schermo dei caratteri alfanumerici e grafici ha bisogno di un puntatore che gli dica dove è situata la memoria di schermo.

Questo puntatore, di cui abbiamo già discusso, è locato in 648 ed il suo valore è uguale all'inizio della memoria di schermo diviso per 256. In condizioni normali esso vale 4 (1024/256).

Siccome il nostro nuovo schermo inizierà da 50176 (49152+1024), dovremo immettere il valore 196 (50176/256) nel vettore 648:

POKE 648,196

Lo spostamento della mappa dei caratteri

Contrariamente a quanto molti si aspettavano, il punto cruciale di tutto l'articolo è posto in ultimo, in quanto è inutile spostare il vettore contenuto in 648 che indirizza la mappa dei caratteri se non si sono compiute in precedenza le necessarie operazioni di inizializzazione, pena uno sgradevole e caotico effetto grafico dovuto al mancato posizionamento dei parametri di visualizzazione.

Una parte del registro situato in 53272, (e precisamente i bit di posizione 3-1), è adibita all'indirizzamento della mappa dei caratteri da cui il Vic II prende l'immagine da abbinare ad ognuno dei 256 codici di schermo.

Come avevamo già fatto notare, lo spazio occupato da una mappa dei caratteri completa è di 2K (2048 byte).

Potremo supporre, dato che con tre bit si possono rappresentare 8 valori (=2 elevato alla 3), che il puntatore possa indirizzare 16/2=8 mappe di caratteri separate nell'ambito dei 16K a sua disposizione.

In effetti il registro si comporta così e la tabella N.3 relativa alle corrispondenze di inizio mappa dei caratteri—valore di 53272 ce lo mostra.

Dobbiamo porre particolare attenzione agli indirizzi d'inizio, in quanto indicano valori che valgono solo per il Vic II in condizioni normali. Siccome abbiamo spostato il suo raggio d'azione a partire da 49152, dovremo aggiungere questo valore a quelli indicati nella terza colonna della tabella.

Per provocare il cambiamento di mappa dovremo digitare la seguente istruzione:

POKE53272,(PEEK(53272)AND241)OR(2*I)

dove I rappresenta il valore indicato nella prima colonna della tabella relativo all'indirizzo desiderato.

Dato che dobbiamo spostare la mappa dei caratteri a partire da 51200, dovremo considerare l'indirizzo 51200—49152=2048, ricavarne il valore del numero di mappa (1) e, infine, eseguire:

POKE53272,(PEEK(53272)AND241)OR2

Prima di eseguire la suddetta istruzione, è necessario abilitare il computer alla visualizzazione dei caratteri minuscoli con:

PRINTCHR$(14)

Il gioco: Funghi e carote

Per passare in modo meno traumatico possibile dalla teoria alla pratica, abbiamo pensato di scrivere un giochino in Basic facile—facile (forse banale) in modo che ognuno possa capirne il funzionamento.

Comunque, forniremo un commento in modo da facilitare ulteriormente un'eventuale modifica del programma, che è stato concepito così scarno proprio per stimolare la fantasia e la creatività dell'utente.

Linee 10—80: Queste linee sono il nucleo del gioco: infatti esse provvedono al movimento della bocca e alla rivelazione dell'urto con un fungo (codice 128) o con una carota (codice 129) comportandosi di conseguenza.
Linee 500—530: Quando il gioco termina (ciò è indicato dalla variabile U che diventa negativa) la linea 50 passa il controllo alla linea 500 che provvede a stampare il tempo trascorso dall'inizio del gioco.

Linea 800: E' un ciclo di ritardo chiamato dalla subroutine che gestisce l'animazione della bocca che mastica la carota.

Linee 900—920: Quando la bocca incontra una carota (linea 40) viene chiamata questa subroutine, che provvede a stampare successivamente le immagini del terzo carattere modificato (codice 130) e dello spazio inverso simulando la chiusura e la riapertura della bocca.

Linee 1000—1080: Queste linee costituiscono la parte che prepara la grafica e inizializza le variabili utilizzate; per offrire una trattazione esauriente di questa parte di programma, lo commenteremo riga per riga:
1000: assegna il colore nero al bordo e allo schermo;

1010: assegna il repeat a tutti i tasti;
1020: legge la routine in linguaggio macchina che trasferisce i caratteri dalla ROM alla RAM che inizia da 51200 e la esegue;
1030: legge i caratteri ridefiniti e li sostituisce al posto delle " @ ","A","B" in reverse (codici 128, 129, 130);
1040: depone nel puntatore 648 il valore 196, sceglie i caratteri minuscoli, il colore grigio3 e blocca il cambiamento del set di caratteri mediante la contemporanea pressione di SHIFT e del logo Commodore;
1050: seleziona il banco di memoria video che parte da 49152;
1060: indirizza il vettore 53272 in modo che punti alla nuova mappa dei caratteri sita in 51200;
1070: inizializza le variabili usate;
1080: stampa le istruzioni, azzera la variabile cronometro TI$ e inizia il gioco.

Linee 1200—1220: Sono i codici della routine in L.M.

Linee 1230—1250: Questi valori costituiscono l'immagine dei nuovi tre caratteri.

Linee 2000—2400: Questa parte di programma provvede alla stampa delle istruzioni di gioco.

Attenzione: interrompendo il gioco con la pressione contemporanea dei tasti Run/Stop e Restore, si "ritorna" allo schermo originario (1024—2023) che differisce da quello utilizzato nel gioco.

Per ripristinare completamente il banco di memoria che si presenta all'accensione del calcolatore, digitate "alla cieca" GOTO10000: tutto tornerà "normale".

```
100 REM  LISTATO 4
110 REM  GIOCO DIMOSTRATIVO SUL
    LA
120 REM  RIDEFINIZIONE DEI CARA
    TTERI
130 REM  BY PASQUALE D'ANDRETI
140 REM  RIARDO (CE)
150 :
160 GOTO 470
170 GET A$:IF A$="S" AND X>0 TH
    EN X=X-1
180 IF A$="D" AND X<39 THEN X=X
    +1
190 POKE I+X,D
200 S=PEEK(Q+X):IF S=G THEN U=U
    +1:GOSUB 400
210 IF S=B THEN U=U-5:IF U<0 TH
    EN 310
220 POKE P+RND(.)*A,B
230 IF RND(.)>.7 THEN POKE P+RN
    D(.)*A,G
240 FOR Y=1 TO 10:NEXT:PRINT:GO
    TO 170
250 :
260 REM  LE LINEE CHE VANNO DA
    170 A 240 COSTITUISCONO
270 REM  IL LOOP PRINCIPALE DEL
     GIOCO.
280 REM  QUELLE DA 310 A 340 IN
    DICANO  IL  PUNTEGGIO
290 REM  RELATIVO AL TEMPO TRAS
    CORSO  DALL'  INIZIO.
300 :
310 A$=TI$
320 A$=LEFT$(A$,2)+" ORE, "+MID
    $(A$,3,2)+" MINUTI, "+RIGHT
    $(A$,2)+ " SECONDI."
330 PRINTCHR$(147)
340 PRINT"HAI RESISTITO PER":PR
    INTA$:END
350 FOR Y=1 TO 90:NEXT:RETURN
360 :
370 REM  SUBROUTINE CHE GESTISC
    E L' ANIMAZIONE
```

```
380 REM  DELLA BOCCA.
390 :
400 POKE Q+X,V:POKE I+X,C2:GOSU
    B 350
410 POKE I+X,C1:GOSUB 350:POKE
    I+X,C2:GOSUB 350
420 POKE I+X,O:RETURN
430 :
440 REM  INIZIALIZZAZIONE DELLE
     VARIABILI, DEI
450 REM  CARATTERI E DELLO SCHE
    RMO.
460 :
470 POKE 53280,0:POKE 53281,0
480 POKE 650,128
490 FOR Y=49152 TO 49202:READ A
    :POKE Y,A:NEXT:SYS49152
500 FOR Y=52224 TO 52247:READ A
    :POKE Y,A:NEXT
510 POKE 648,196:PRINTCHR$(147)
    CHR$(14)CHR$(8)CHR$(155)
520 POKE 56576,PEEK(56576) AND
    252
530 POKE 53272,(PEEK(53272) AND
     240) OR 2:CLR
540 B=128:G=129:I=50176:Q=I+40:
    X=0:P=I+960:U=50:V=32:C1=16
    0:C2=130:A=40:O=218
550 GOSUB 740:FOR Y=1 TO 25:PRI
    NT:NEXT:TI$="000000":GOTO 1
    70
560 :
570 REM  ROUTINE IN LINGUAGGIO
      MACCHINA  CHE
580 REM  RICOPIA I CARATTERI DA
    LLA ROM ALLA RAM
590 REM  LOCATA IN 51200.
600 :
610 DATA 173,14,220,41,254,141,
    14,220,165,1,41,251,133,1,1
    60,7
620 DATA 162,0,189,0,208,157,0,
    200,202,208,247,238,20,192,
    238,23
630 DATA 192,136,208,236,165,1,
    9,4,133,1,173,14,220,9,1,14
    1,14,220,96
640 :
650 REM  BYTES CHE DEFINISCONO
     I TRE CARATTERI
660 REM  PROGRAMMABILI.
670 :
680 DATA 0,24,126,255,255,24,24
    ,24
690 DATA 1,3,6,12,60,120,240,96
700 DATA 255,255,231,195,195,23
    1,255,255
710 :
720 REM  ISTRUZIONI
730 :
740 PRINTCHR$(147)CHR$(155)"IN
    QUESTO GIOCO DOVRAI GUIDARE
     LA BOCCA"
750 PRINT"IN MODO DA MANGIARE L
    E CAROTE (A) ED"
760 PRINT"EVITARE I FUNGHI (@)
    VELENOSI:"
770 PRINT"HAI, ALL'INIZIO, 50 P
    UNTI DI TOLLERANZA"
780 PRINT"AL VELENO, MA OGNI VO
    LTA CHE MANGI UN"
790 PRINT"FUNGO LA TOLLERANZA D
    IMINUISCE DI 5"
800 PRINT"UNITA' MENTRE, PER OG
    NI CAROTA, AUMENTA"
810 PRINT"DI 1 UNITA'."
820 PRINT"IL GIOCO FINISCE SE L
    A TOLLERANZA E'"
830 PRINT"MINORE DI ZERO.":PRIN
    T
840 PRINT"PER SPOSTARE LA BOCCA
     USA I TASTI:"
850 PRINT"S (SINISTRA)  D (DES
    TRA)."
860 PRINT:PRINT"SE, DURANTE IL
    GIOCO, PREMI RUN/STOP E RES
    TORE"
870 PRINT"DIGITA, ALLA CIECA: G
    OTO 10000"
875 PRINTCHR$(18)"PREMI UN TAST
    O PER INIZIARE"
880 GET A$:IF A$="" THEN 880
890 RETURN
900 END
910 REM  RIPRISTINO VALORI ALL'
    ACCENSIONE
10000 POKE 648,4:POKE 56576,199:P
    OKE 53272,21
```

GIOCHI DI GUERRA

Vi è mai capitato di destreggiarvi in un gioco di guerra dalle regole sconosciute? Ecco alcuni utili consigli per sei videogame che vanno per la maggiore.

di Marco Miotti e Carlo Barazzetta

Quelli che descriviamo in questa sezione "speciale" rappresentano quanto di meglio offre il mercato per ciò che riguarda le simulazioni di guerra.

Vi presentiamo infatti i sei giochi che dominano le vette delle classifiche mondiali in questo settore:

1 — Beach head 2: Guida i tuoi uomini contro il tiranno.

2 — Silent service: Simulatore di missioni di guerra a bordo di un sottomarino.

3 — Commando: Conquista l'ultima roccaforte dei nemici.

4 — Dambusters: Immagina di pilotare una fortezza volante contro una postazione del Reich.

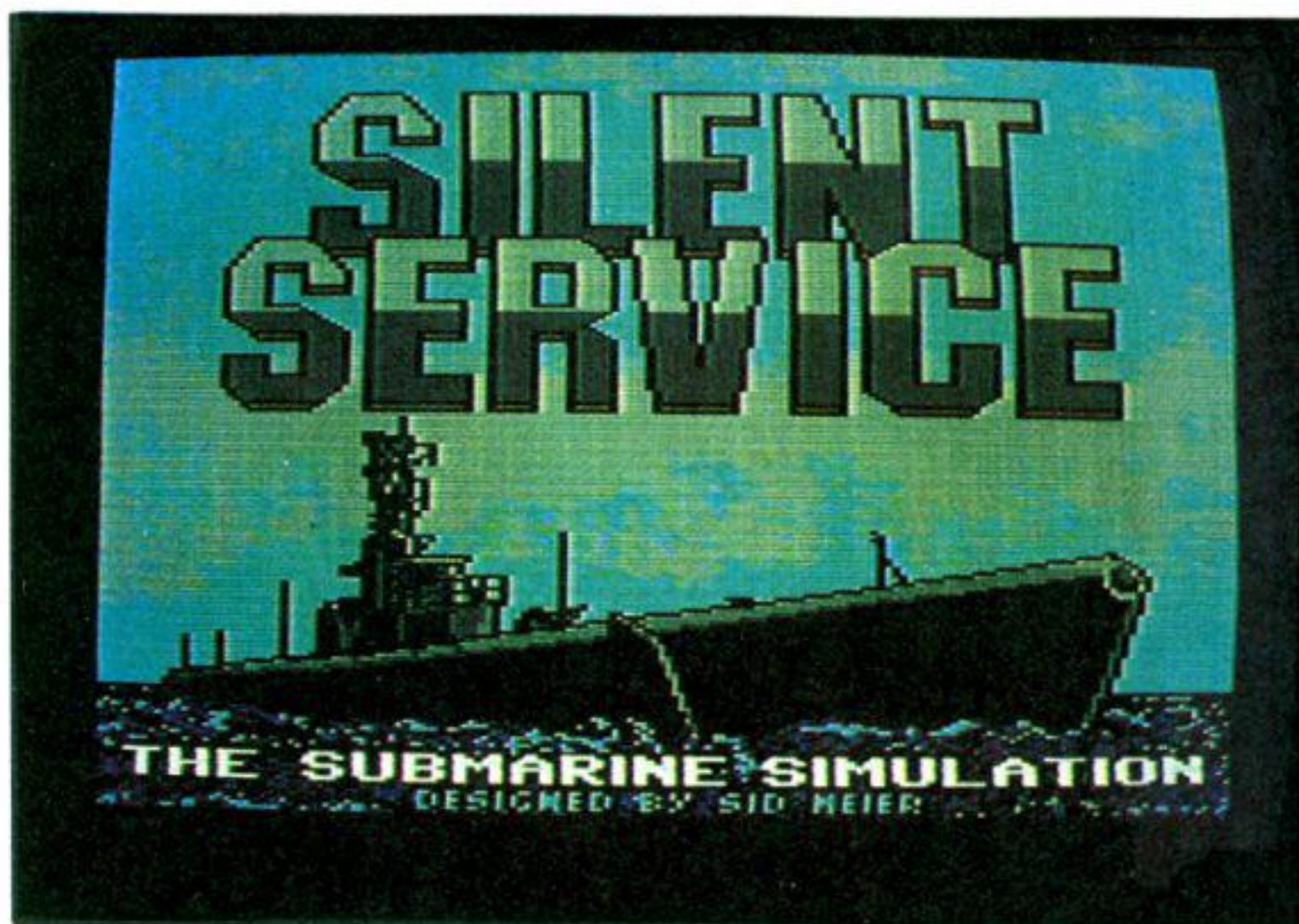
5 — Rambo: tratto dall'omonimo film.

6 — Desert Fox: Combatti contro Rommel tra le dune africane.

Questi articoli sono particolarmente rivolti a coloro che, pur possedendo uno dei giochi descritti, non riescono a trovare le regole ed i comandi per un suo uso corretto.

Silent service

Per gli appassionati di simulazione questo videogame rappresenta la "risposta" navale all'ormai famoso Flight Simulator 2.

di Marco Miotti

Descrivere in dettaglio questo stupendo gioco è assolutamente impossibile dato che è presente una tale varietà di situazioni che lo rendono tra i più difficili ed interessanti del suo genere.

Lo scopo del gioco è comunque immediato da comprendere: compiere un raid nelle acque del Pacifico meridionale, infestato, ovviamente (!), da numerosi convogli giapponesi.

Il periodo storico è quello della seconda guerra mondiale dal 1942 in poi. Le difficoltà che si incontrano sono davvero enormi: basti dire che nell'high—scores l'autore del gioco ha inserito anche la classifica relativa alle più famose missioni dell'ultimo conflitto.

La grafica, non eccezionalmente curata, risulta nel complesso notevole in quanto, unendo tra loro più "moduli" semplici, arriva a formare, volta per volta, schemi eccezionalmente complessi.

Ne è un esempio rilevante la schermata in cui appare il periscopio: ogni elemento dello schermo, "debole" graficamente se considerato a sè stante, crea un effetto inaspettato se considerato insieme agli altri.

Esistono tre diversi livelli tra cui scegliere la difficoltà in cui cimentarsi:

- Scenario
- Livello di difficoltà
- Livello di realtà

Lo scenario sceglie tra:
a — Torpedo/gun practice
b — Convoy actions
c — War patrols

a — Questo scenario ti pone subito in prossimità di quattro navi nemiche che, è inutile dirlo, devi affondare. E' la fase più semplice del videogame perchè consente di aquisire pratica prima di partecipare ad un'azione bellica vera e propria.

b — Nel secondo scenario è previsto l'assalto ad un convoglio nemico formato da navi che trasportano truppe, navi cisterne, eccetera.

c — Qui è previsto un intero pattugliamento del Pacifico che può durare fino a due mesi (tempo, ovviamente, relativo...). Per evitare la monotonia degli scenari privi di eventi particolari l'autore del gioco ha previsto il pattugliamento solo sulla carta geografica (visibile sullo schermo), dove il sottomarino è identificato tramite un punto nero tra le isole. Il trascorrere del tempo è individuato dal colore del bordo dello schermo che da blu chiaro (giorno) passa periodicamente a blu scuro (notte).

Il livello di difficoltà indica al computer il modo di considerare la tua esperienza.

I livelli di difficoltà tra cui scegliere sono quattro:

a — midshipman: E' il livello più basso, generalmente utilizzato dai principianti, ed influisce pochissimo sull'astuzia dei nemici.
b — liutenant: luogotenente. Di livello superiore al precedente, deve essere utilizzato da persone esperte. L'astuzia del nemico raddoppia.
c — Commander: questo livello è riservato solo a coloro che si sentono di operare ad un livello... storico. L'astuzia del computer, tuo avversario, è ancora raddoppiata rispetto alla precedente.
d — Captain: è un livello del tipo... "Non ci riuscirai mai" e rispecchia fedelmente l'estrema tensione che, di norma, accompagnava i capitani delle navi durante le loro missioni.

Il livello di realismo è inversamente proporzionale alle facilitazioni che si avranno durante la missione:

1 — Visibilità limitata.
2 — Zig Zag dei convogli.
3 — Siluri difettosi.
4 — Riparazioni solo nei porti.
5 — Distruttori esperti.
6 — Ricerca dei convogli.
7 — Calcolo dell'angolo di impatto.

Quando una di queste opzioni è scelta, il livello di difficoltà generale cresce restando in un range compreso tra 0 e 9. Tutte queste caratteristiche saranno annotate e verranno poi memorizzate nell'high—scores.

Stazioni di battaglia

— Torre di comando

La torre di comando è la stazione di battaglia primaria, dove cioè si decidono tutte le mosse della battaglia che verrà combattuta. Da questa posizione, infatti, il capitano (i cui movimenti sono comandati dal joystick), indicherà la stazione scelta. Quando ci si trova nella torre di comando il gioco è fermo, consentendo così un'ampia riflessione sulle mosse future.

Per accedere alla torre di comando è

sufficiente premere il fire del joystick in qualunque momento del gioco; si seleziona quindi la stazione di battaglia desiderata posizionandovi l'omino e premendo contemporaneamente il fire.

— Mappe di navigazione

Questa stazione consente l'esplorazione di quella parte del Pacifico interessata dall'azione; lo sfarfallio del bordo dello schermo indica il passaggio dal giorno alla notte, mentre quando si incrocia un convoglio nemico il bordo dello schermo assume il colore rosso. A questo punto, premendo il fire, si ingaggerà la battaglia.

— Mappe di combattimento

Queste sono mappe particolareggiate della zona in cui ci si trova e servono per orientarsi nelle azioni di guerra. E' possibile scorgere ogni movimento delle navi nemiche, come sullo schermo di un radar.

— Ponte

Questa stazione mostra l'esterno del

sottomarino con un impressionante effetto realistico. E' anche possibile (purchè ci si trovi in emersione!), guardare il... panorama.

— Periscopio

Viene utilizzato nelle azioni di guerra per centrare l'obiettivo in un collimatore in modo da calcolare la rotta dei siluri. Quando ci si trova in immersione viene utilizzato anche per dirigersi.

— Strumentazione

Indica lo stato dei motori e delle parti meccaniche da guerra; particolarmente utile è l'indicatore di profondità che consente di esaminare l'effetto delle azioni di guerra in immediata vicinanza del nemico.

— Rapporto danni

Lo schermo indica la natura di tutti i danni riportati dal sottomarino.

Un breve giudizio

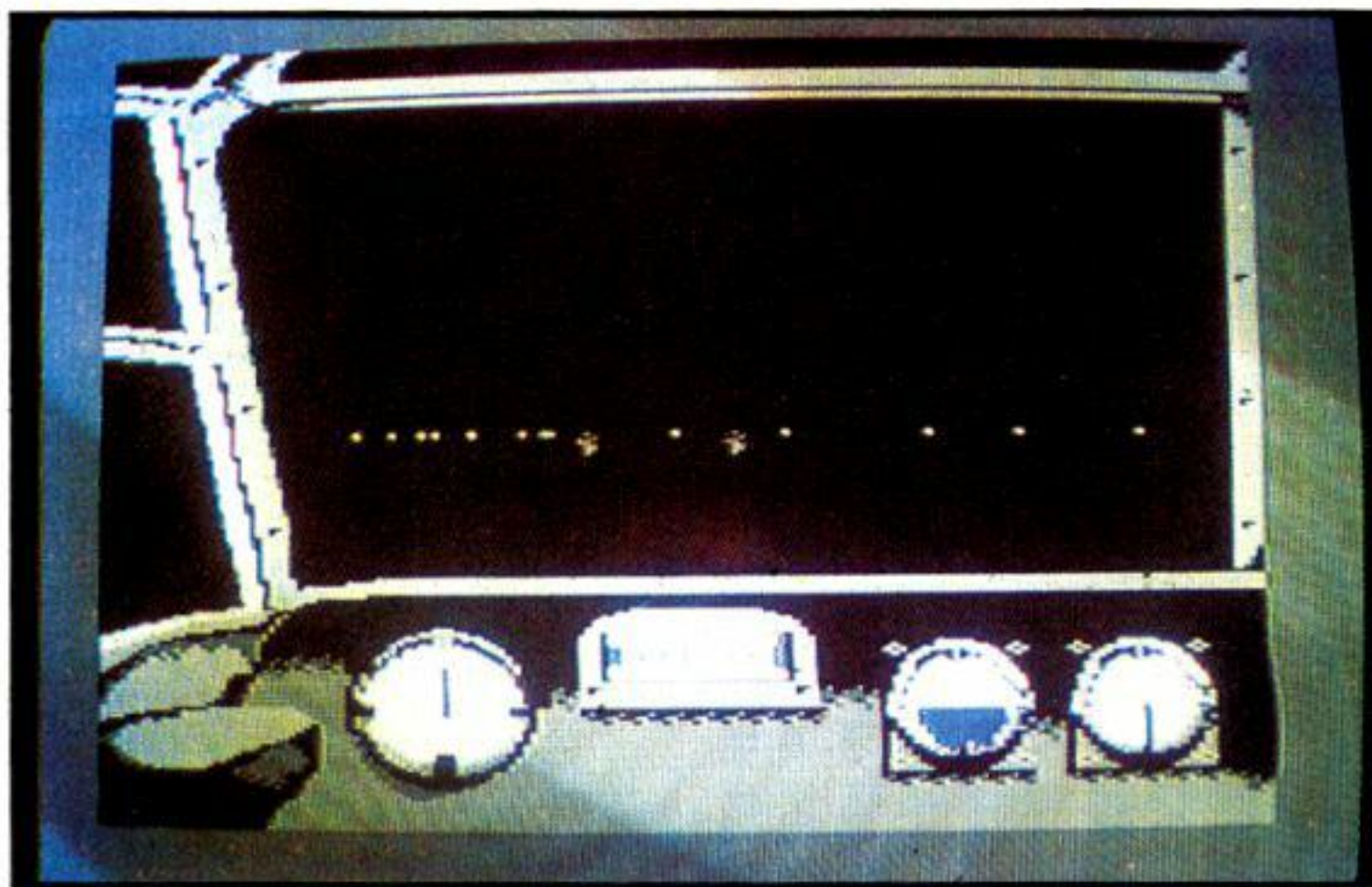
Il realismo del gioco, davvero ammirevole, deve essere considerato insieme con le possibilità di manovra del sottomarino che rendono difficile ed affascinante la missione.

Particolarmente interessante è la possibilità di viaggiare realmente attraverso il Pacifico ammirandone la bellezza e la varietà dei paesaggi. L'unico inconveniente è costituito dal fatto che, in questo caso, si rischia di viaggiare per mesi interi senza incontrare nessuno.

Non potendo elencare tutte le possibilità e le strategie che il videogame consente, ci limitiamo a ricordarne i comandi principali:
Velocita: 0—4
Inverte i motori: R
Periscopio su/giù: P
Siluri: T
Cannone: G
Destra: Tasto cursore destra/sinistra.
Sinistra: cursore su/giù.
Il Pacifico vi attende.
Buon viaggio!

Dambusters

Una simulazione incredibilmente realistica di una missione aerea inglese sul territorio della Germania nazista. Riuscirete nell'intento?

di Marco Miotti

E' l'alba del 15 maggio 1945.

Da un aeroporto dell'Inghilterra del sud decolla una delle più moderne fortezze volanti: il Lancaster

Direzione: Nord della Germania.

Obiettivo: Abbattere la più importante delle dighe presenti sul territorio nemico, in modo da non permetterne l'utilizzo alle forze naziste.

La missione non è delle più semplici: occorrono infatti particolari capacità di comando, in quanto è necessario dirigere l'aereo attraverso le trappole preparate dalle forze nemiche.

Occorre fare attenzione per evitare i palloni frenati, per "seminare" o, eventualmente, abbattere i caccia notturni tedeschi, a non farsi scorgere dalle luci delle fotoelettriche, perchè un Lancaster è un facile bersaglio per la contraerea. Devi, insomma, pilotare l'aereo magistralmente, ricorrendo a tutta l'esperienza che hai accumulato nella lunga carriera di pilota.

Gli appassionati di simulazione avranno sicuramente drizzato le orecchie: quando si tratta di pilotare un apparecchio, nei loro animi si accende la luce della curiosità e quanto più è realistico il gioco, maggiormente sono stimolati a saperne di più.

L'effetto realistico è davvero notevole, fin troppo. Il controllo dell' aereo risulta qui meno difficile di quello del Flight simulator 2 (per chi non conoscesse questo fantastico simulatore di volo, diciamo semplicemente che ricalca fedelmente la realtà ed è particolarmente indicato per colui che, miserello, non possiede un aereoplano...).

L'aereo, generalmente, non presenta particolari problemi di guida, eccetto forse per quel che riguarda la regolazione "fine del volo". Il manuale indica infatti le seguenti specifiche di volo per poter abbattere la diga:

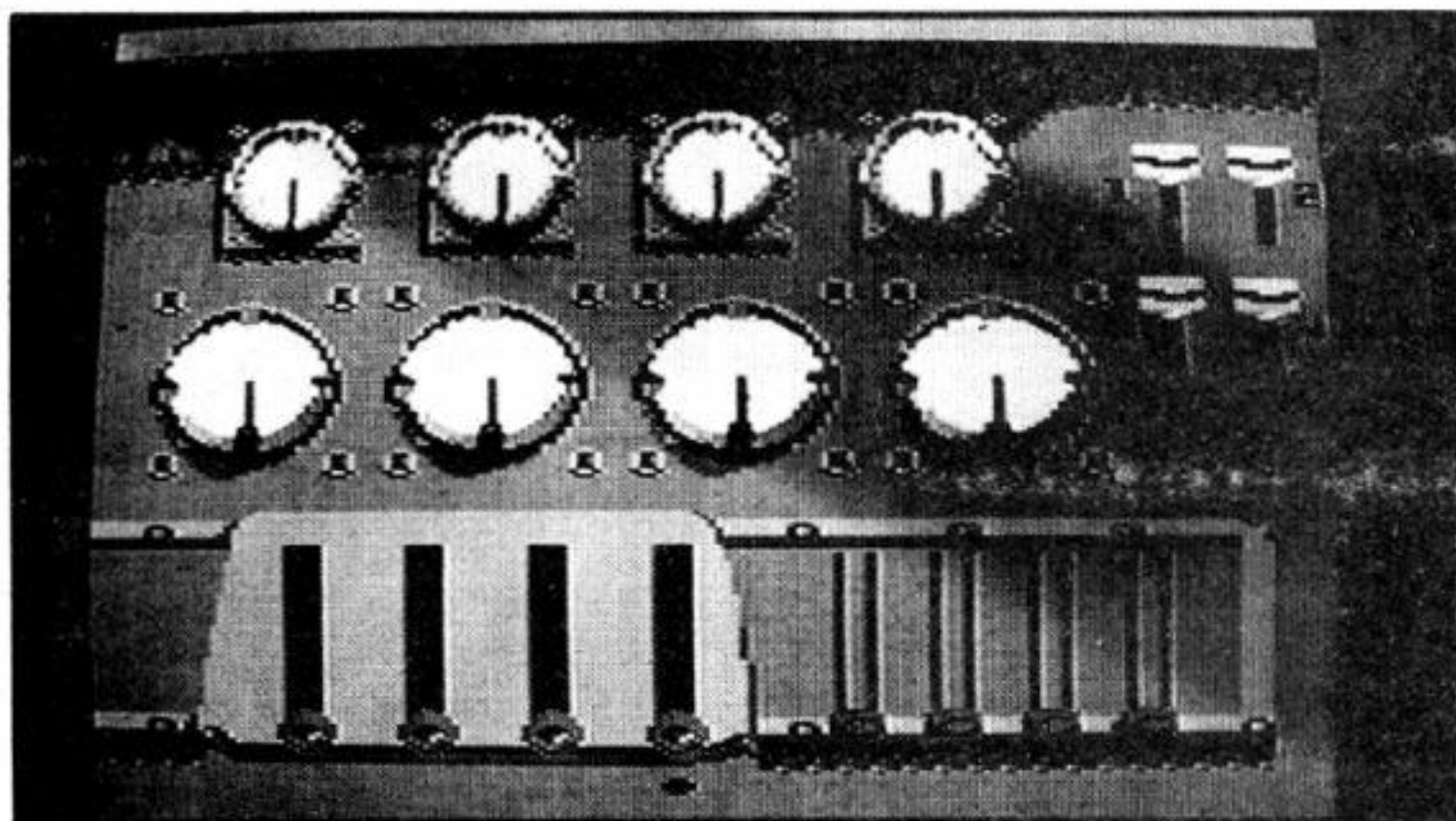
- 232 mph (miglia orarie) di velocità.
- 60 piedi di altezza.
- Distanza: deve essere esattamente di 800 yard.

Assicuriamo che raggiungere questi parametri è un compito fin troppo difficile, ma siccome San Progettista pensa per noi, ci è data l'opportunità di accedere al gioco step by step (in una prossima serie di articoli vi daremo la traduzione) cominciando con:

Dam approach

Il vostro C—64 si assume il gravoso compito di portarvi sul lago artificiale, trattenuto dalla diga che dovete bombardare.

Il vostro compito si riduce (per modo di dire) a regolare i parametri prima descritti ed a sganciare la bomba.

Non avrete complicazioni di nessun genere: nessun pallone frenato, luci di ricerca o apparecchi ME 110 (caccia notturni).

Flight lieutenant

La tua missione comincia sulla Manica e devi coraggiosamente guidare il Lancaster attraverso le sopracitate insidie (non sarà facile!), tracciare la rotta e, ovviamente, seguirla. Arrivato alla diga non hai che da colpirla dato che, ormai, sei espertissimo nel bombardarla (Dam approach).

Squadron leader

Con questa fase la missione è davvero completa: parti dall'Inghilterra, vai a colpire e torni. Devi solo imparare anche a decollare ed atterrare.
Nell'opzione Squadron leader c'e anche il quadro di controllo di:
a) Flaps
b) Carrello
c) Trimmer di regolazione del timone di direzione

Passiamo ora ad una descrizione dei quadri di controllo del volo, selezionabili premendo i tasti da 1 a 8.

I quadri di controllo

1) Schermo del pilota: si controlla il volo dell' aereo e si ha una completa visione del cielo immediatamente di fronte al pilota. Oltre all'altimetro sono raffigurati il compasso direzionale, l'orizzonte artificiale (che dà la situazione dell'orizzonte rispetto all'aereo) ed infine l'indicatore della velocità dell'aria.

Inutile dire che il joystick controlla i movimenti dell'aereo. Il tasto Fire non ha nessun effetto, in quanto il pilota di un bombardiere non deve sparare.

2) Mitragliatore frontale: ha il compito di abbattere i vari ostacoli di natura bellica che si parano davanti al Lancaster: i movimenti sono relativamente semplici e consistono esclusivamente nello spostare il collimatore in prossimità del bersaglio e sparare, ovviamente, utilizzando il Joystick.

Al mitragliere viene impartito anche il comando di rotazione, puntamento e sgancio della bomba. Questa, ovviamente, sarà lanciata non appena l'aereo sarà alla distanza voluta, misurata a vista dal collimatore speciale posto sul fondo dello schermo. Anche in questo caso occorre premere il Fire per sganciare la bomba.

3) Mitragliere di coda: questa postazione provvede ad eliminare gli oggetti che si avvicinano all'aereo dalla parte posteriore: i controlli sono simili a quelli previsti per il mitragliere frontale, fatta eccezione per il fatto che il mitragliere di coda non controlla nessuna bomba.

4) Secondo quadro del mitragliere frontale: qui sono immediatamente visibili due controlli: l'interruttore che comanda la rotazione delle bombe e il comando che accende le luci di sincronizzazione.

Per accendere le luci occorre portare l'aereo sotto i 100 piedi, così che le luci, se puntate sull' acqua, danno un esatto riferimento per l'altezza; infatti se l'altezza dell'aereo fosse superiore a 60 piedi allora le luci formerebbero due cerchi sull'acqua; se l'altezza fosse invece inferiore le luci si incrocerebbero tra loro. L'esatta altezza è data dai due cerchi di luce che combaciano perfettamente (attenzione; questo è l'ultimo dei parametri da regolare, in quanto una riduzione o un aumento del regime dei motori influenza l'altezza del velivolo). Il mitragliere frontale in questo quadro può controllare direttamente l'altezza dell'aereo, consentendogli così una più precisa regolazione dei parametri.

5) Schermo del navigatore: qui vengono programmati tutti gli aspetti del volo: rotta, velocità, obiettivo ecc.

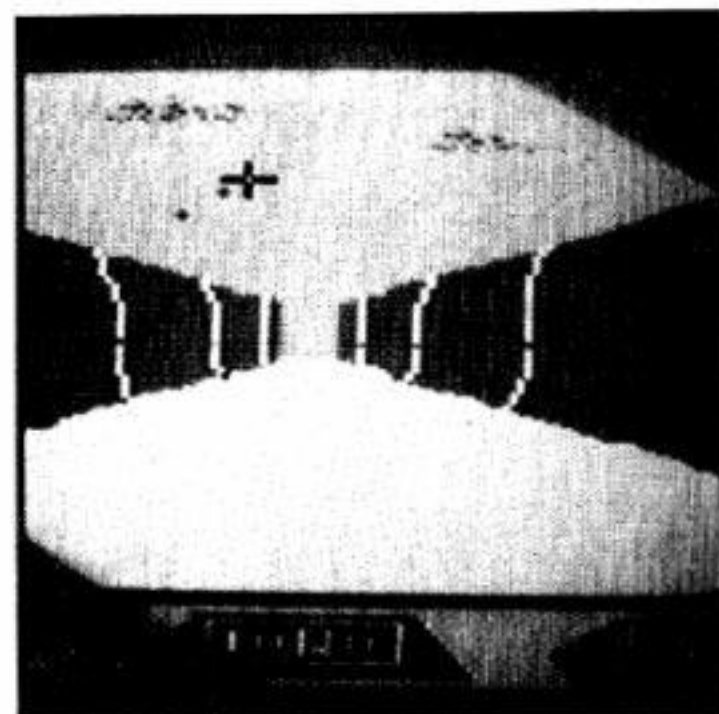
Il navigatore ha il diretto controllo della direzione dell'aereo che è segnata su un compasso direzionale posto alla sommità dello schermo. La posizione dell'aereo è chiaramente individuata da un aeroplanino che si muove attraverso il video.

6) Schermo dell'ingegnere di bordo: qui si trovano le quattro leve di comando dell'aereo, quelle che controllano il rapporto giri—motore giri—elica, gli interruttori che azionano gli estintori (posti su ogni motore) e gli indicatori dello stato dei motori.

7) Secondo schermo dell'ingegnere: è lo schermo utilizzabile soltanto nelle azioni complete e controlla i flaps, il carrello di atterraggio e il trimmer del timone di direzione.

8) Rapporto danni: in questo schermo vengono presentati i danni prodotti dalla contraerea o dai caccia ME 110. E' compito del giocatore stabilire se, in base al più recente rapporto danni, è il caso di continuare, o meno, la missione.

Desert Fox

Gli episodi dell'ultimo conflitto sono di ispirazione per questo gioco di simulazione di notevole interesse.

di Carlo Barazzetta

Nell'inverno del 1941 il generale tedesco Rommel, soprannovinato "la volpe del deserto" (in inglese "desert fox"), comandava le truppe tedesche nel nord Africa e ingaggiò, contro gli inglesi, la "battaglia della Marmarica".

In questo videogame il giocatore deve guidare le forze inglesi per sconfiggere l'esercito tedesco, come accadde realmente nel novembre del 1941 quando Rommel fu costretto a ritirarsi da questa zona nordafricana.

Il gioco si presenta con un menù di opzioni. Prima di affrontare il gioco vero e proprio avete la possibilità di allenarvi nei diversi tipi di combattimento:

Stuka Attack (attacco aereo)
Tank Duel (duello fra carriarmati)
Mine Field (campo minato)
Ambush Attack (imboscata)
Convoy Attack (attacco al convoglio)

Quando vi sentite pronti potete cominciare la battaglia scegliendo, fra cinque livelli di difficoltà crescente, la vostra campagna militare. In seguito appare una cartina geografica che riporta fedelmente la zona in cui avvennero i combattimenti.

Un piccolo carro armato rappresenta la vostra posizione, mentre la svastica rappresenta la posizione di Rommel.

E' ovviamente necessario comandare le vostre truppe con abilità spostandole lungo il territorio. Il vostro scopo è quello di giungere in aiuto ai presidi inglesi (simboleggiati dalle bandierine) minacciati dalle truppe tedesche.

Per spostare le truppe di rinforzo, possedete una radio con la quale potete sondare le zone intorno a voi in modo da sapere se andate incontro alle forze nemiche. In questa fase il gioco utilizza la sintesi vocale per pronunciare i messaggi radio: quando, per esempio, pronuncia "Stuka" vi informa che vi state dirigendo verso una squadriglia aerea.

Mediante un ulteriore comando (Zoom) potete osservare la quantità di truppe che possedete sui diversi fronti.

Se perdete le vostre truppe in un presidio, senza che siate riusciti a raggiungerlo con altre truppe di rinforzo, il gioco si conclude infelicemente.

Potete comunque giocare un'ultima carta (quando vi accorgete di avere pochi uomini a disposizione), inviando un aereo sul luogo.

La battaglia si concluderà vittoriosamente, costringendo Rommel ad alzare bandiera bianca, solo se riuscirete ad imporvi sull'intero territorio.

Il gioco si dimostra piacevole in tutte le fasi e mette a dura prova le capacità tattiche del giocatore. Non sono da trascurare, comunque, la prontezza di riflessi e il "colpo d'occhio" necessari in quasi tutti i videogames di guerra.

Come giudizio critico diremo che i disegni non sono molto curati pur se compaiono numerose scenografie. Ottima la sintesi vocale, ma si sente la mancanza di una colonna sonora. La simulazione è sostanzialmente buona, specie considerando i movimenti, realizzati anche in prospettiva. La giocabilità, ad ogni buon conto, non è frustrante dato che è possibile selezionare diverse difficoltà.

Comunicato

La Systems Editoriale S.p.a

In seguito
alle recenti notizie degli organi di stampa nazionale

Ritenendo
che parte delle suddette notizie siano lesive dell'immagine dell'azienda stessa

Precisa quanto segue:

1/ Le apparecchiature in dotazione della Systems Editoriale e dei suoi collaboratori sono modelli del tutto normali e devono ritenersi praticamente identici a quelli posti in commercio nei punti di vendita specializzati.

2/ Le prove di trasmissione e ricezione dati via computer effettuate dalla Systems Editoriale non sono mai state realizzate con scopi illeciti.

3/ Il presunto collegamento con la banca dati militare della NATO non è mai stato realizzato.

4/ La Systems Editoriale non ha mai inviato intenzionalmente il segnale di allarme atomico di secondo grado, pervenuto in una base militare del Mediterraneo nel periodo dal 4 all'11 febbraio u.s.

5/ La conferma del decollo dei bombardieri B-52 dotati di cariche nucleari, per una non meglio identificata missione di attacco, non è mai partita dalla sede ufficiale della Systems Editoriale.

6/ Pur ammettendo la possibilità di interferenze in trasmissione dati, la Systems Editoriale dubita fortemente di essere stata la causa della presunta violazione della rete informatica militare, specialmente nel periodo citato 4/11 febbraio '85.

La Systems Editoriale, pertanto,

Diffida

chiunque insista, per mezzo di notizie false e tendenziose ad attribuire alla stessa Systems Editoriale responsabilità che le sono del tutto estranee.

Milano 1/4/86

Commando

Esser da soli, in un ambiente ostile, con una missione pericolosa da compiere. Nessuno, finora, ci è riuscito. E tu?

di Marco Miotti

L'ordine da eseguire è tassativo: espugnare ad ogni costo la roccaforte avversaria nascosta nella jungla. Solo a te potevano affidare la responsabilità della missione, solo tu sei capace di cose sovrumane, di combattere da solo, contro cento uomini, per scoprire poi che ce ne sono altri cento...

Il realismo utilizzato nel gioco riesce però, ad attanagliare l'attenzione del giocatore che, magari, si aspetta, alla fine, di vedere scorrere i titoli in coda, come se avesse assistito ad un autentico film.

Niente di tutto questo, ovviamente, perchè si tratta solo di un videogioco.

La grafica stupendamente curata, che offre una varietà notevole di scenari, e la musica di sottofondo eseguita con gran cura, hanno quel certo non so che...

E visto che non lo sappiamo, passiamo alla descrizione del gioco, premettendo che nessuno di noi è riuscito ad arrivare alla conclusione della missione.

Il gioco

Sei nella Jungla: non puoi pretendere che le guardie non si accorgano della tua presenza, ed infatti ecco che arrivano, e devi sparare per difenderti.

Questo, a proposito, è uno dei tanti giochi in cui è sempre meglio sparare che restare inattivi: più spari più punti guadagni. I nemici arrivano da tutte le parti, ma è possibile colpirli sia con un potente mitragliatore da azionare (guarda un po') premendo il tasto Fire del joystick, sia lanciando delle bombe, mediante la pressione della barra spaziatrice.

E se resto senza bombe, come faccio? Calma, fermi tutti! Esiste la possibilità di rifornirsi durante il "viaggio", derubando i nemici, appropriandosi cioè delle casse di bombe abbandonate lungo la strada (le potrai identificare dal fatto che lampeggiano: meglio di così!). Contrariamente alle bombe, i colpi del mitragliatore non si esauriscono mai, per cui niente paura: non sarete mai costretti ad una lotta corpo a corpo per sopravvivere.

E' molto facile farsi distrarre dalla stupenda grafica, tanto da non riuscire più a giocare come si deve. Ricordati però che tu sei l'eroe senza macchia e senza paura del momento, per cui lascia stare paesaggi e disco—music e cerca di abbattere quanti più avversari è possibile, a raccattare il maggior numero di bombe e a liberare i numerosi prigionieri scortati da due perfidi armigeri che dovrai uccidere.

Ovviamente gli avversari non sono stupidi e, di norma, a nessuno piace farsi colpire, per cui occhio ai colpi che ti fioccano intorno ed alle bombe che descrivono pericolose parabole sopra la tua testa.

A proposito: vi consigliamo di conservare le bombe per utilizzarle solo nel passaggio dei punti più difficili, come ad esempio sotto il ponte.

Ma cosa vedo? (boing!) un grosso portone! Allora sono riuscito a superare le insidie della jungla nera e ad arrivare sano e salvo all'ingresso della roccaforte.

Qui occorre destrezza. Occhio al sergente, consente di guadagnare un bel mucchio di punti.

Liberata tutta la zona, si passa al livello successivo, dove occorrerà attraversare numerose insidie nell'ultimo lembo di jungla che ti separa dal campo nemico.

E' ora necessario passare attraverso i colpi di bazooka del nemico, dannatamente precisi.

Gli scenari successivi sono un susseguirsi di vari campi di battaglia (ponti, aereoporti ed altro) dove la dinamicità delle azioni è assicurata da un fine e preciso scrolling che consente la continuità degli scenari: il tutto corredato da una fantastica musica di sottofondo.

Al contrario di altri giochi di guerra, questo non rappresenta una simulazione, ma deve esser considerato come un vero e proprio game d'azione in cui la prontezza di riflessi è determinante per la completa riuscita della missione.

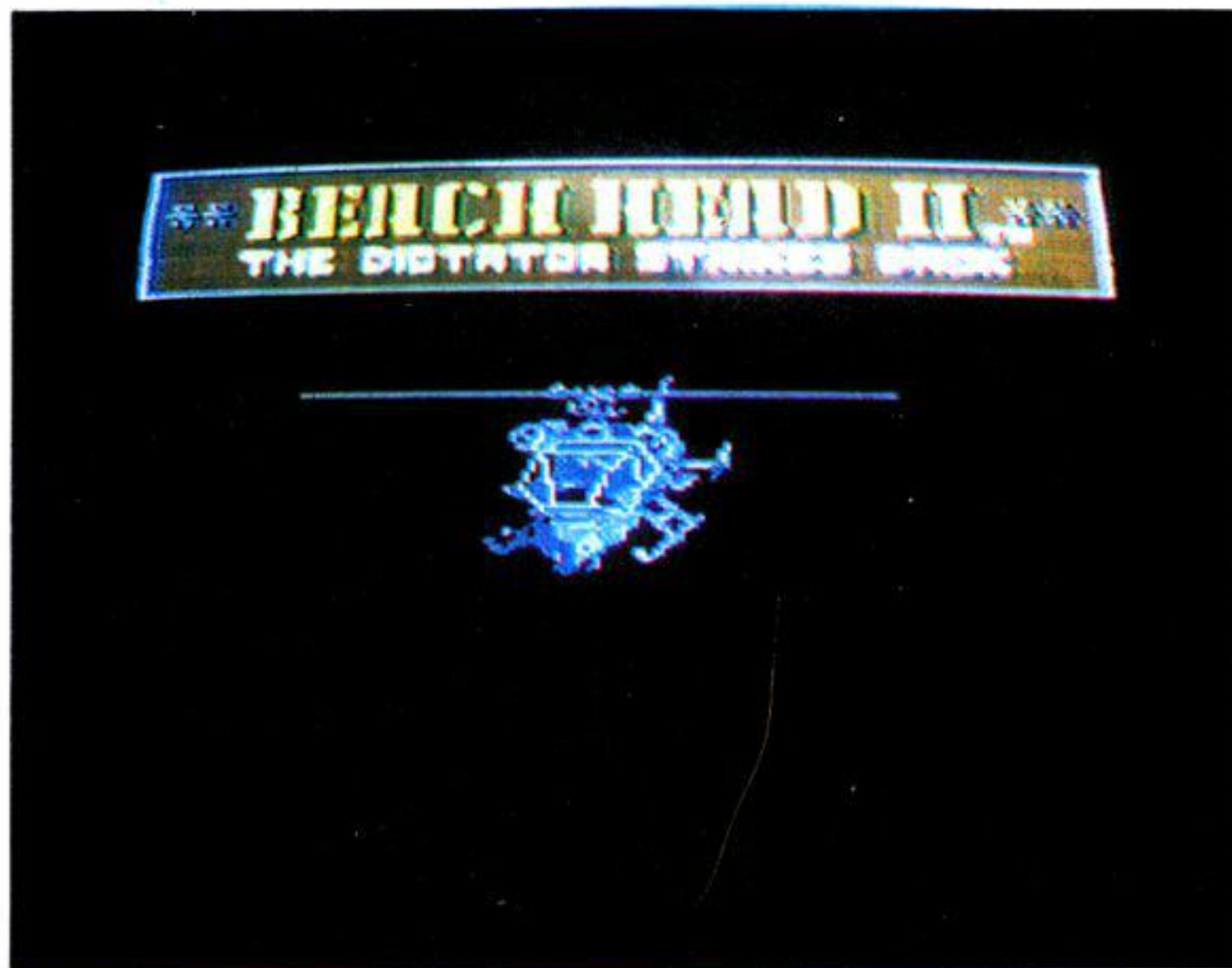
I controlli sono molto semplici: il joystick dirige l'omino in una qualsiasi delle direzioni. Da rilevare, però, che (come in tutte le azioni suicida) qui non è consentito tornare indietro: occorre vincere per poterlo fare. Il tasto Fire controlla il mitragliatore, mentre la barra spaziatrice lancia le bombe antiuomo tipo "ananas".

Non esistono particolari strategie per la conduzione ottimale del gioco, se non quella di stare con gli occhi aperti e indovinare subito gli spostamenti da effettuare.

E se proprio non riuscite a portare a termine la missione limitatevi a gustare la grafica e la musica del gioco.

Beach Head II

I colpi fioccano attorno a te come la neve, i prigionieri invocano il tuo aiuto e alla fine, se sei ancora vivo, devi uccidere il perfido dittatore.

di Mauro Miotti

Una mitragliatrice nemica spara contro i tuoi compagni cercando di impedire ad ogni costo la vostra avanzata.

E tocca a te: hai già superato i primi due muri posti sul campo di battaglia e ti accingi a compiere una missione suicida mentre uno dei tuoi compagni viene mandato avanti per distogliere l'attenzione dei nemici. Devi portarti davanti alla torretta nemica e, stando bene attento ad evitare i colpi che ti fischiano intorno, tirare una granata verso la postazione avversaria.

Comincia così uno dei più emozionanti giochi di guerra presenti attualmente sul mercato software per Commodore 64. Continuazione dell'ormai famoso Beach Head (prima versione) dove lo scopo era di espugnare il bunker del dittatore, questo gioco si presenta con la didascalia: The dictator strikes back.

Lo scopo del gioco è quindi di eliminare una volta per tutte il sadico dittatore e liberare gli ostaggi trattenuti nel campo di prigionia.

Per fare ciò occorre superare quattro diverse fasi, una più rischiosa dell'altra:

Prima fase: Attacco

L'abilità del giocatore è qui messa a dura prova dalle raffiche esplose dalla postazione comandata dal dittatore. Un elicottero, comandato dal giocatore, catapulta uomini paracadutandoli sul campo. L'elicottero deve posizionarsi non solo in corrispondenza di uno dei muri protettivi, ma anche ad un'altezza sufficiente per consentire l'apertura dei paracadute.

Una volta scaricati tutti i soldati (sia che essi siano arrivati a destinazione o... meno) l'elicottero si allontana ed il giocatore passa al controllo diretto degli uomini.

Una voce, ben sintetizzata, pronuncia, infatti "Follow me men" e le estremità dei muri cominciano a lampeggiare, indicando a voi (e purtroppo anche al dittatore) da quale parte uscirà un soldato. A questo punto aspettate che lampeggi la parte desiderata e poi premete il Fire del joystick muovendolo contemporaneamente verso destra o verso sinistra per dirigere il vostro uomo verso uno dei muri (il secondo gruppo di muri, ovviamente).

Se tutto va bene, il vostro omino riuscirà a mettersi in salvo dietro il secondo muro, altrimenti... i suoni emessi dal computer non lasciano spazio a dubbi!

Quando tutti gli uomini dal primo muro saranno passati al riparo del secondo (oppure saranno stati uccisi) allora il gioco si sposta al muro successivo.

Le modalità di gioco, in questa seconda fase, sono simili a quelle precedenti. C'è solo da notare che, schiacciando il tasto Fire, si manda verso la mitragliatrice un uomo—diversivo (che non verrà conteggiato nel numero degli uomini caduti sul campo). E' necessario, inoltre, muovere il joystick per comandare l'omino—commando che dovrà dirigersi al centro davanti alla mitragliatrice (evitandone i colpi micidiali) e gettare la granata.

Se il colpo andrà a segno, la torretta verrà distrutta. L'omino deve, in seguito, porsi in salvo al di sotto della torretta, posto decisamente sicuro, al di fuori del tiro del nemico.

Nel caso la torretta venga distrutta, sarà rimpiazzata da un'altra simile non appena il milite raggiungerà la base della stessa.

I movimenti del dittatore sono molto semplici e siccome ci si può calare sia nel personaggio del dittatore sia in quella dell'alleato (così vengono chiamati gli attaccanti dagli autori del videogame) sarebbe ingiusto non descrivere i movimenti possibili della torretta di fuoco.

Il Joystick controlla completamente i movimenti della torretta (ovviamente attraverso la control port diversa da quella che controllava gli alleati). Il pulsante di sparo controlla quindi le raffiche dirette verso i poveri soldatini. E' comunque possibile conoscere la direzione verso cui sparerà la mitragliatrice: basta seguire la traiettoria dei primi colpi e dirigere il proprio uomo immaginando la traiettoria di quelli successivi.

Occorre notare che la torretta si muove più velocemente se non vi sono sparatorie in corso.

Dopo che tutti gli uomini saranno passati davanti alla torretta (sia che essi abbiano lanciato o meno la granata) si passa alla seconda fase.

Seconda fase: recupero dei prigionieri

Gli alleati si sono impadroniti della torretta e l'hanno rivolta verso il campo di prigionia del dittatore.

Mossi da un barlume di speranza, i prigionieri cominciano ad attraversare il cortile e sarà compito del giocatore difenderli dai sadici attacchi delle guardie che, rimaste senza armi da fuoco, cercheranno di gettar in testa ai fuggitivi ogni sorta di oggetti, oppure di crear trappole mortali accompagnate da risate sataniche.

A complicar la vita del povero e malconcio prigioniero, concorrono inoltre un carro armato, che tenterà di schiacciarlo, ed un autoblindo equipaggiato con un mortaio che, non appena giunge in prossimità del malcapitato lo toglie repentinamente di torno.

La parte più complicata spetta ora al dittatore: mentre i movimenti del carro armato e delle autoblindo sono automatici, le guardie vanno spostate come segue:

— Portando la leva verso il basso, si scelgono le guardie di terra e le si fanno muovere. Per depositare una trappola occorre premere il Fire del Joystick.

— Spingendo la leva in alto si muove la guardia posta alla sommità del muro;

premendo il Fire si provvede a lanciare sul prigioniero un oggetto.

— Tirando la leva a destra e spingendola in alto, oppure tirandola in basso, si anima, rispettivamente, la guardia sul muro oppure quella di terra posizionata a destra.

— Portando invece la leva a sinistra si ottiene, come effetto, il movimento delle guardie a sinistra.

Il compito degli alleati è quello di proteggere la lentissima avanzata del prigioniero attraverso il cortile del campo. Il joystick controlla infatti la mitragliatrice che può colpire uno qualsiasi degli oggetti in movimento (anche il prigioniero... occhio, quindi!). Per accelerare la velocità di attraversamento del cortile occorre sparare ad una delle tre porte poste sul muro di fronte.

Il compito del dittatore è, in questa fase, molto più gravoso di quello assolto dagli alleati nella prima parte del gioco.

L'elicottero, che aveva portato i soldati per l'azione militare, non è fuggito, ma aspetta il loro ritorno per dare inizio alla terza fase.

Terza fase: La fuga degli ostaggi

Riservato esclusivamente agli alleati, questo quadro presenta difficoltà non irrilevanti ed una stretta analogia con il penultimo quadro del precedente Beach Head.

L'elicottero carica i prigionieri e tenta di portarli fuori dal territorio nemico cercando di evitare carri armati, missili e altre trappole presenti lungo il percorso.

L'elicottero è guidato dal Joystick nel seguente modo:

— Destra: esegue una virata verso destra.

— Sinistra: esegue una virata verso sinistra.

— Alto: esegue una picchiata.

— Basso: cabra. (N.B: Sono questi i reali controlli di un vero velivolo. Giocando, invece, si è portati istintivamente a spingere la cloche per abbassarsi, sbagliando così manovra).

— Fire: lancio di un missile.

Vi sono diversi livelli di difficoltà per questo quadro, livelli che verranno selezionati volta per volta dal computer in base ad un criterio sconosciuto. Per questo motivo non sarà possibile sapere a quale livello si sta giocando fin quando non si incontreranno le prime difficoltà.

Si hanno a disposizione fino a sei elicotteri (che corrispondono al massimo grado di difficoltà) per cui il consiglio che diamo è quello di non caricar tutti i prigionieri sul primo elicottero, ma di distribuirli equamente tenendo presente che per ogni elicottero arrivato a destinazione il numero degli elicotteri rimanenti può non essere quello che ci si aspetta. Questo perchè, in effetti, vi sono tre soli elicotteri, con due sole possibilità di percorso per ciascuno. Se, quindi, si giunge al termine della missione al primo tentativo, gli elicotteri rimasti non saranno cinque bensì quattro.

Quando tutti i prigionieri si saranno

allontanati dal campo comincerà, finalmente, la lotta finale.

Quarta fase: Resa dei conti

La base del malvagio dittatore sorge sulle rovine di un antichissimo monastero dimenticato da centinaia di anni.

Questo monastero si snoda attraverso un complicatissimo labirinto di caverne che coprono distanze enormi. Il dittatore, che conosce l'esistenza del labirinto, cerca quindi scampo nelle viscere della terra.

Il giocatore, cioè tu, scoperta la fuga, invia i suoi uomini nelle caverne alla ricerca del sadico dittatore. Ma sarai proprio tu a trovarlo, rifugiato nella volta centrale, nel posto più inaccessibile del monastero. Comincia così l'affascinante battaglia tra bene e male.

Armati solo di grossi e precisi punteruoli cercherete, tu da una parte e lui dall'altra di colpirvi e di uccidervi: uno solo sopravviverà e sarà il vincitore della battaglia finale.

I controlli sono identici ma simmetrici: il dittatore infatti si trova alla sinistra dello schermo, mentre il capo degli alleati alla destra, entrambi separati da un lago sotterraneo.

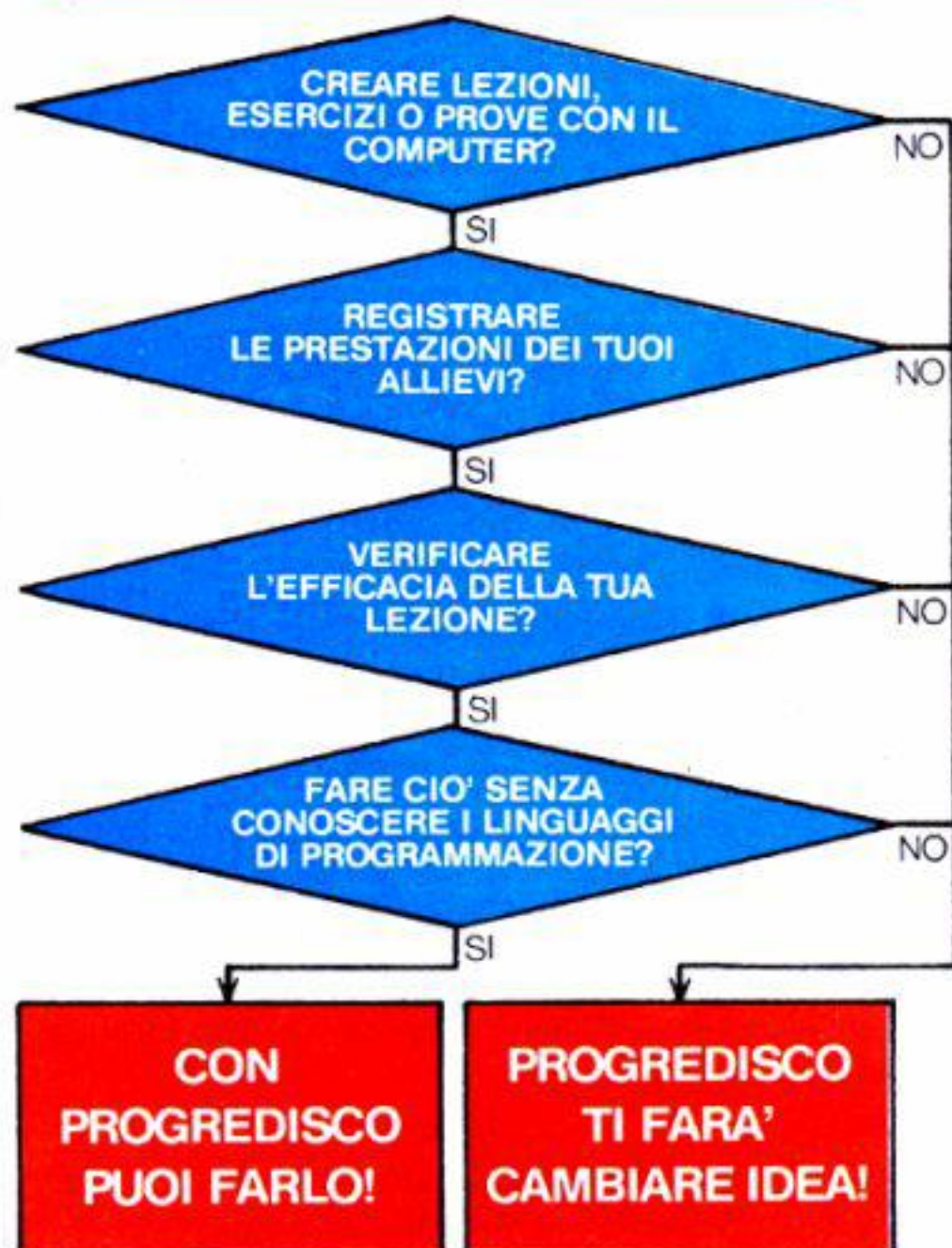
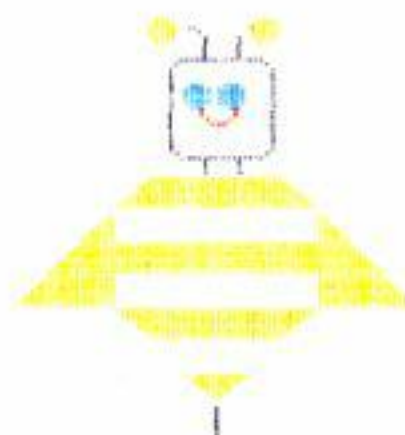
Muovendo il Joystick su e giù si muove l'omino lungo la passerella; schiacciando il Fire si lancia un coltello dritto davanti; se, insieme al Fire, si porta la leva nella posizione alto—sinistra o basso—sinistra per l'alleato oppure nelle posizioni alto—destra, basso—destra per il dittatore, si lancia il coltello nelle rispettive direzioni.

L'uomo che verrà colpito per quattro volte cadrà nel lago sottostante e darà partita vinta all'altro. Questo duello si ripeterà per dieci volte. In seguito verranno sommati i punteggi parziali ed alla fine risulterà vincitore colui che avrà realizzato il punteggio maggiore.

Abbiamo descritto, con dovizia di particolari, la battaglia finale, ma... la guerra?

Tutti i punteggi realizzati da entrambe le parti verranno a questo punto sommati e solo alla fine si saprà se la vittoria sarà stata totale oppure... di Pirro.

La grafica e la dinamica del gioco sono realmente fantastiche. La creatività del giocatore è stimolata dalle situazioni e dagli eventuali imprevisti che rendono questo gioco una tra i più richiesti ed impegnativi.

Occorre non poca abilità per guidare gli uomini attraverso la lotta sia che ci si trovi nei panni degli alleati sia in quelli del dittatore; l'unica strategia consigliata è quella di provare a giocare e non arrendersi di fronte alle prime, inevitabili difficoltà.

Rambo

di Marco Miotti

Chi... sarà in grado di guidarlo nella jungla del Vietnam alla ricerca del campo di prigionia che tiene ancora segregati i suoi compagni? Tu, ovviamente!

Scopo del gioco è quello di liberare tutti i compagni di Rambo cercando di evitare le sentinelle vietnamite e tutti i soldati che spuntano come funghi, compresi quelli americani che ti hanno abbandonato.

Il campo di battaglia è davvero vasto: orientarsi non è sempre facile ma sicuramente riuscirai a districarti tra le insidie della jungla fino a raggiungere lo scopo che ti eri prefisso.

Dapprima dovrai recuperare il fucile, che avevi abbandonato in un tempio per poterti muovere più agevolmente nell'esplorazione. In seguito dovrai svolgere il compito più difficile: liberare i tuoi compagni. Dovrai infatti localizzare il campo di prigionia e tagliare le corde che legano il tuo amico ad una croce.

Svelto come un fulmine dovrai ora impadronirti dell'elicottero con cui andare alla ricerca degli altri compagni di sventura.

La storia non finisce ovviamente qui, ma per motivi di spazio siamo costretti a troncarla: quanto ci sarebbe da parlare su un'avventura di questo genere!

In questo gioco, decisamente dinamico, Rambo fa rivivere al giocatore le vicende del taciturno "berretto verde". Il tutto unito ad un miscuglio di ottima grafica, fantastici suoni che nulla tolgono, tuttavia, all'attenzione del giocatore ed ad un'infinita varietà di azioni. Da un simile minestrone non poteva che nascere qualcosa di veramente unico nel suo genere, anche se un paragone fra due prodotti è praticamente impossibile, in quanto nulla di ciò che si trova sul mercato software per C—64 presenta affinità con altri suoi simili. Ogni programma si può quindi sempre definire unico.

Per controllare Rambo occorre muovere il joystick in una delle quattro direzioni. Contrariamente a Commando, qui le azioni si svolgono in qualsiasi direzione, senza seguire un percorso fisso: la vastità del luogo è davvero notevole.

Il tasto Fire ovviamente serve per sparare, mentre l'arma da utilizzare si sceglie premendo la barra spaziatrice. Una volta recuperato il mitragliatore si può scegliere tra:

- Coltello (modello Rambo).
- Mitragliatore (modello mega—plus arci e chi più ne ha più ne metta).
- Bomba anti—uomo (la classica granata tipo "ananas").
- Bomba di grande potenza per aprirsi il varco tra le piante.

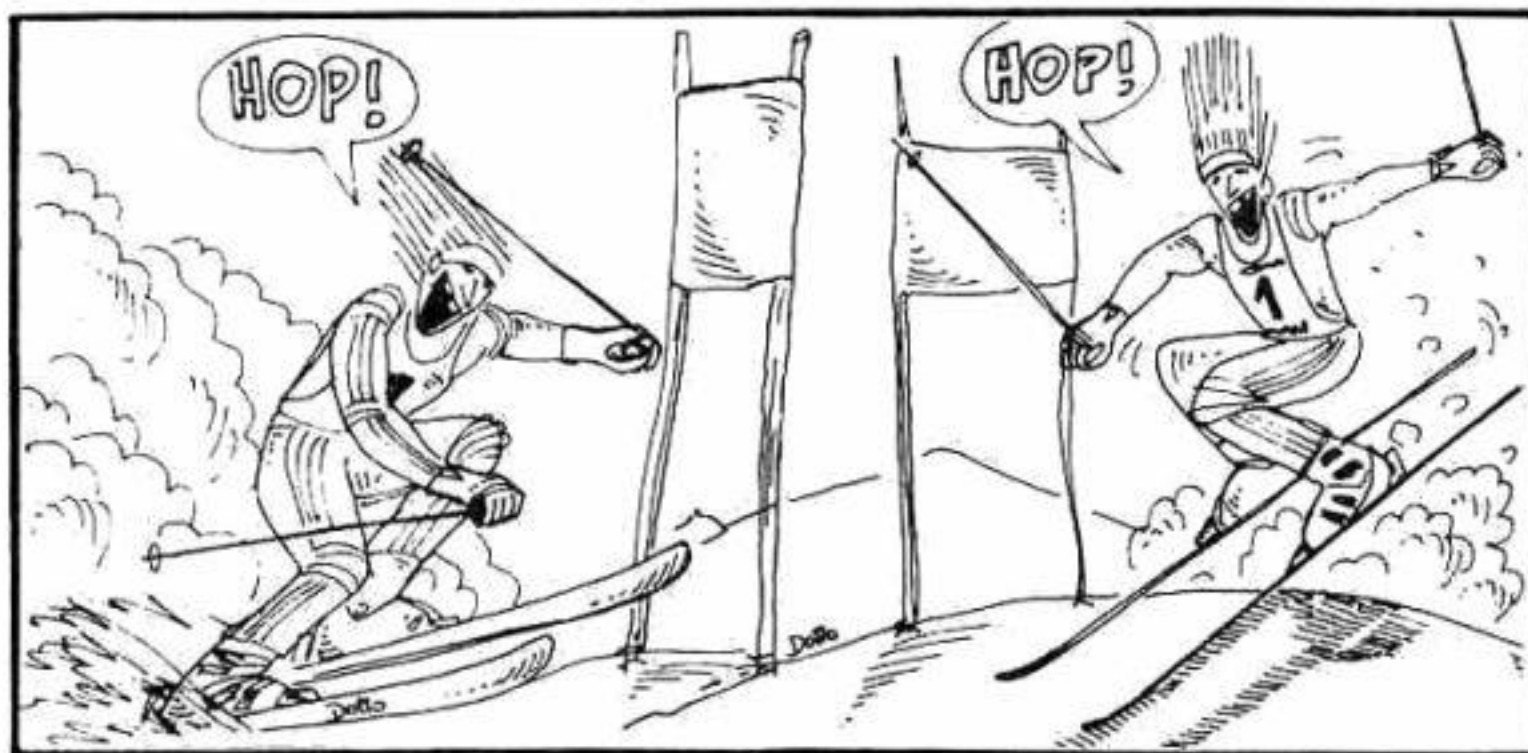
Anche in questo gioco, come in tanti altri, conta prevalentemente l'abilità del giocatore unitamente a una gran dose di pazienza.

Rimandiamo tutto alle vostre prossime avventure nella jungla, nelle profondità del mare oppure su una fortezza volante.

Ma ovviamente non finisce qui... da qualche parte qualcuno ha bisogno del tuo coraggio per altre missioni.

Una gimcana tra studio e gioco

Vediamo che cosa avviene quando il C-16 incontra il comando Escape... e approfittiamone per giocare.

di Claudio Baiocchi

Lo scrolling del video è una delle fasi per le quali deve passare ogni gioco di movimento.

Se il programma non è realizzato in Linguaggio Macchina, la velocità del gioco risulterà inaccettabile. C'è tuttavia una comoda scappatoia: usare le routine di sistema che eseguono, appunto, lo scrolling. Su C-16 e Plus/4 si hanno, per esempio, a disposizione le routine che gestiscono il tasto Escape, ma su ogni computer il semplice comando PRINT può servire allo scopo.

Progettiamo un videogame

Supponiamo di voler costruire un gioco di tipo "gimcana": la nostra posizione sullo schermo è un pallino che possiamo manovrare azionando il joystick o premendo tasti opportuni. Scopo del gioco è evitare alcuni ostacoli che, generati casualmente, ci vengono incontro.

La realizzazione del programma dovrà passare attraverso fasi diverse, in cui "P" indica la nostra posizione sullo schermo. La prima fase è necessaria se non vogliamo che il pallino lasci una scia:

- cancellare il pallino nella posizione P;
- eseguire uno scrolling dello schermo;
- interrogare la tastiera o il joystick e, se necessario, modificare il valore di P;
- leggere il contenuto della posizione P;
- scrivere il pallino in posizione P;
- se la posizione P era occupata, generare effetti visivi e/o sonori;
- generare ostacoli casuali e tornare alla prima fase.

Se ci accontentiamo di una gimkana sciistica (in cui la nostra posizione è in alto sullo schermo e gli ostacoli nascono dal basso e ci vengono incontro salendo) la prima fase non è necessaria e la realizzazione del programma è banale.

La seconda fase, che è la più delicata, si ottiene semplicemente portando il cursore a fine schermo ed eseguendo un PRINT!

Su C-16 e Plus/4, grazie ai comandi strutturati DO e LOOP, bastano quattro righe di programma: si veda il listato 1 in cui i movimenti si ottengono manovrando i tasti D (destra) e S (sinistra). Modificando in linea 1 i valori di "L" (lunghezza della scia) e di "D" (coefficiente di difficoltà) si può constatare come già in poche linee di programma si possono ottenere giochi interessanti e veloci. Una versione espansa, commentata e adattabile anche a Vic-20 e C-64 è fornita nel listato 2.

Se invece vogliamo costruire una gimcana automobilistica (la nostra posizione è in basso e gli ostacoli ci vengono incontro scendendo) la seconda fase del programma diventa irrealizzabile in Basic. Per avere un gioco sufficientemente veloce dobbiamo inevitabilmente lavorare in LM.

Eppure, su C-16 e Plus/4, la funzione ESC "W" sarebbe il toccasana. Peccato che il manuale non spiega come inserirla in un programma.

Potremmo, tramite il programma Monitor incorporato, spulciare nella ROM alla ricerca della routine corrispondente; poi da Basic, fare una SYS a tale routine.

C'è, fortunatamente, una strada più breve: basta inserire, nel punto in cui si vuole lo scrolling verso il basso, il comando Basic:

```
PRINT CHR$(27) CHR$(87)
```

e il gioco è fatto.

Il listato 3 (in cui i tasti da manovrare sono ancora D ed S e il numero 3 in linea 60 può essere modificato per variare la difficoltà del gioco) ne mostra una realiz-

zazione, sempre in quattro righe di programma.

Come complicarsi la vita

Una volta scoperto il trucco, può venir voglia di fare cose più interessanti; ad esempio il listato 4 permette di gestire varie finestre sul video. Il programma 5, inoltre, ridefinisce alcuni tasti funzionali. Più precisamente al tasto F8 è assegnata la pulizia dello schermo dalla posizione del cursore in poi; ai tasti F1 e F4 sono assegnate mansioni utili per lavorare in LM: passati in Monitor, tenendo premuto il tasto F1 si ottiene un disassemblaggio non stop (il disassemblaggio cessa rilasciando il tasto stesso e ricomincia se lo premiamo ancora).

Il tasto F4 ha un ruolo analogo, solo che lavora all'indietro, scrivendo ovviamente nella prima riga dello schermo e facendo slittare all'ingiù le righe successive.

Informazioni utili

Concludiamo con qualche informazione utile a chi vuole saperne di più sul tasto Escape: la tabella degli indirizzi di inizio delle routine relative, "parte" dalla locazione ROM $DE1A (parte bassa poi parte alta per ESC A; parte bassa poi parte alta per ESC B e così via).

Aumentando di 1 tali valori si ottiene l'indirizzo cercato (cui fare una SYS o dal quale iniziare a disassemblare, eccetera).

ALLARME ROSSO
(per i principianti)

Nel digitare righe di programma basic che contengono istruzioni DATA, è piuttosto facile incorrere in errori di digitazione. Supponiamo che un'ipotetica linea basic numerata con 1200 debba contenere i tre valori: 123, 456, 789. Ecco alcuni esempi di errori più frequentemente commessi:

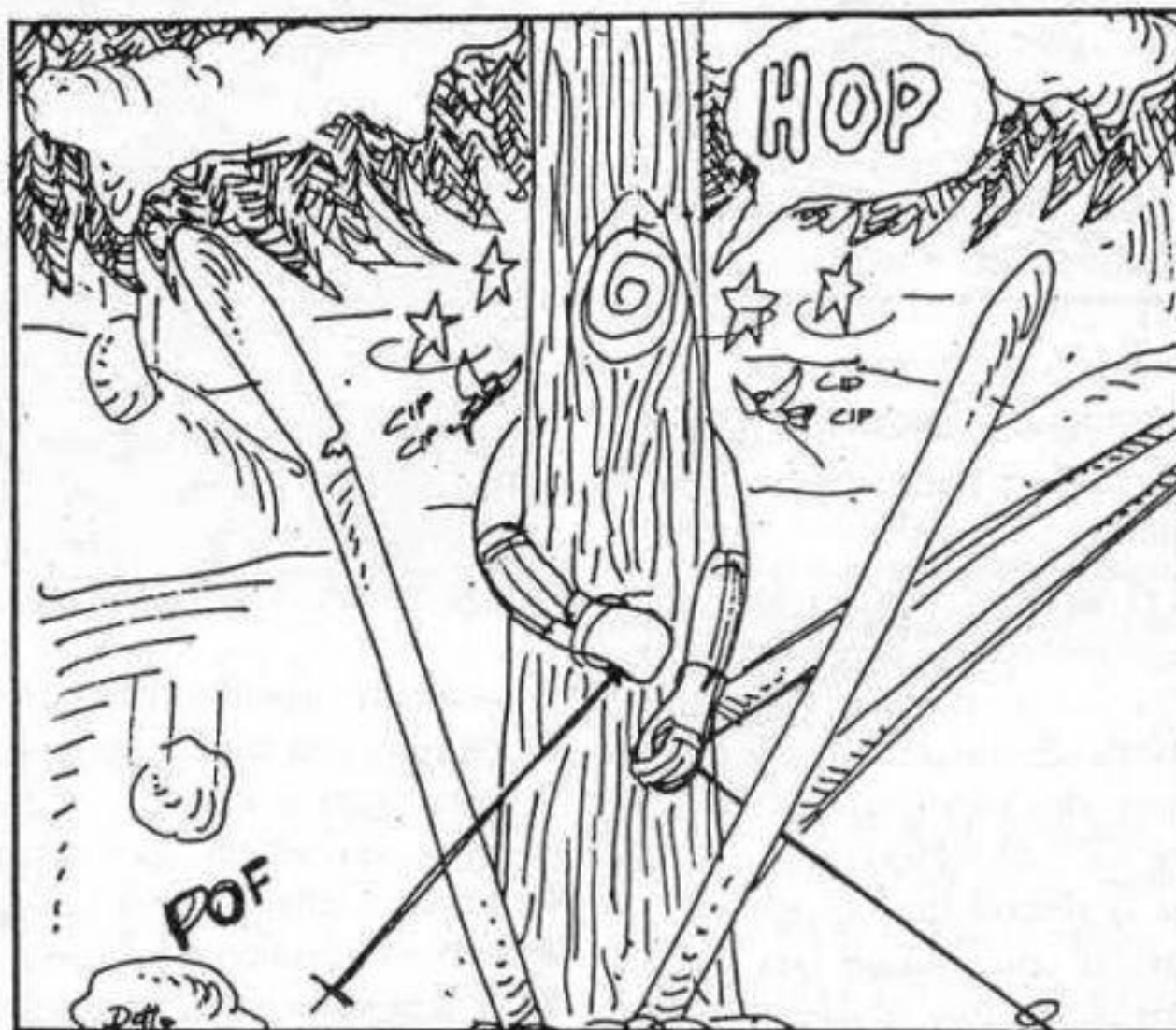
1200 DATA,123,456,789

C'è una virgola dopo la parola "DATA". I dati letti dal computer sono, in questo caso, quattro: 0, 123, 456, 789. Se, infatti, non figura alcun carattere dopo l'istruzione DATA, automaticamente viene assunto il valore nullo (0).

1200 DATA 123,456,789,

In questo caso, dopo il 789, il computer, grazie alla presenza della virgola erroneamente inserita, "crede che ci sia un altro valore e, non trovandolo, lo assume come nullo (0).

1200 DATA 1234,56,789

La virgola è posizionata male, vale a dire dopo il carattere "4" e non dopo il carattere "3". Il computer non può sapere se il valore esatto è 123 oppure 1234 e individuare un errore, in questo caso, risulta piuttosto laborioso.

```
10 REM MINI GIOCO
20 REM PER C-16
30 REM IN QUATTRO RIGHE
40 :
50 REM TASTI "S" E "D"
60 :
70 D=3:T=2038:A=19:SCNCLR:DO:POKEA+4032,81:FORX=1TOD
80 PRINTSPC(39*RND(1))"*"CHR$(19);:NEXT:P=PEEK(T)
90 A=A-(P=18)+(P=13):A=(A>39)*(A-39)-A*(A>0)
100 PRINTCHR$(27)"I":LOOPWHILEPEEK(A+4032)=32:RUN
```

Alcune delle funzioni di ESC sono d'altronde gestibili direttamente con delle POKE.

Ad esempio, ESC "L" ed ESC "M" equivalgono a pokare nella locazione decimale 2025 il valore 0 e 128 rispettivamente. ESC "A" ed ESC "C" equivalgono a pokare nella locazione decimale 2026 il valore 255 e 0 rispettivamente. Infine le memorie da 2021 a 2024 contengono ultima e prima riga, prima e ultima colonna, della finestra attiva nel video.

```
100 REM SLALOM:
110 REM MINI - GIOCO
120 REM PER QUALSIASI
130 REM COMMODORE
140 :
150 REM DI CLAUDIO BAIOCCHI
160 :
170 PRINTCHR$(147)"1- C/64":PRINT"2- C/16 PLUS/4":PRINT"3- VIC/20"
180 GETA$:IFA$=""THEN180
190 IFA$="1"THENTP=197:S=1024:C=39
200 IFA$="2"THENTP=2038:S=3072:C=39
210 IFA$="3"THENTP=197:S=7680:C=21
220 IFVAL(A$)<1ORVAL(A$)>3THEN180
230 REM TP=TASTO PREMUTO. C=N.COLONNE-1.SU VIC20 FARE C=21
240 REM S=INIZIO SCHERMO.SU C64 S=1024. SU VIC20 INESPANSO S=7680
250 PRINTCHR$(147):PRINT:PRINTCHR$(18)," S  L  A  L  O  M ":PRINT:PRINT
260 PRINT"PREMI "CHR$(18)"D"CHR$(146)" PER ANDARE A DESTRA":PRINT
270 PRINT"PREMI "CHR$(18)"S"CHR$(146)" PER ANDARE A SINISTRA":PRINT
280 INPUT"DIFFICOLTA' (1-9)";D:IFD<1ORD>9THEND=3
290 PRINT:PRINT"LUNGHEZZA SCIA (INTERO, TRA 0 E.20)"
300 INPUTL%:IFL%<0ORL%>20THENL%=5
310 FORX=1TO9E9:REM SIMULA UN DO--LOOP
320 FORY=1TO30:PRINT:NEXT:REM PULISCE SCHERMO E PORTA CURSORE IN BASSO
330 A=C/2:REM POSIZIONE INIZIALE A META' RIGO
340 FORY=-1TO0:REM SIMULA UN DO--LOOP WHILE
350 PRINT:REM SCROLLING VERSO L'ALTO
360 T=PEEK(TP):A=A-(T=18)+(T=13):REM MODIFICA POSIZIONE SE RICHIESTO
370 A=(A>C)*(A-C)-A*(A>0):REM NON SI DEVE CAMBIARE RIGA: SERVE 0<=A<=C
380 P=A+S+L%*(1+C):REM POSIZIONE SCIATORE
390 Y=PEEK(P)=32:REM Y=-1 SE POSIZIONE LIBERA; Y=0 SE OCCUPATA
400 POKEP,81:REM SCRIVE PALLINO;SU C64 E VIC20 VECCHIE ROM OCCORRE
410 REM CAMBIARE IL COLORE DI SFONDO PER RENDERE VISIBILI LE POKE DI SCHERMO
420 REM O FARE ANCHE UNA POKE IN MAPPA COLORE
430 FORZ=1TOD:PRINTSPC(C*RND(1))"*"CHR$(145):NEXT:REM PRINTA A CASO DEI *
440 NEXT:REM LOOP WHILE Y
450 NEXT
```

```
10 REM MINI SLALOM
20 REM PER C-16
30 :
40 REM TASTI "S" E "D"
50 :
60 L=5:D=3:A=19:FORY=1TO30:PRINT:NEXT:DO:PRINT:T=PEEK(2038)
70 A=A-(T=18)+(T=13):A=(A>39)*(A-39)-A*(A>0)
80 P=A+3072+L*40:Y=PEEK(P)=32:POKEP,81:FORZ=1TOD
90 PRINTSPC(39*RND(1))"*"CHR$(145):NEXT:LOOPWHILEY:RUN
```

```
10 REM ********************************
20 REM *                              *
30 REM *  C R E A   F I N E S T R E   *
40 REM *  (SOLO PER C-16 E PLUS/4)    *
50 REM ********************************
60 :
70 REM FINESTRA 0 = TUTTO SCHERMO; FINESTRE DA 1 A 10 A VOLONTA'
80 REM VOLENDO PIU' DI 10 FINESTRE OCCORRE DIMENSIONARE I VETTORI :
90 REM RI(.) = RIGA INIZIALE
100 REM RF(.) = RIGA FINALE
110 REM CI(.) = COLONNA INIZIALE
120 REM CF(.) = COLONNA FINALE
130 REM  R(.) = RIGA CURSORE
140 REM  C(.) = COLONNA CURSORE
150 REM VARIABILI USATE :
```

```
160 REM AT = # FINESTRA ATTUALE
170 REM  F = # FINESTRA DA CREARE
180 REM RI,RF,CI,CF ESTREMI FINESTRA DA CREARE
190 REM ROUTINE DI SISTEMA USATE :
200 REM SYS 55369 : SCRIVE POSIZIONE CURSORE IN 2035,2036
210 REM SYS 55355 : POSIZIONA CURSORE COME INDICATO IN 2035,2036
220 REM NON PUO' ESSERE USATA INDISCRIMINATAMENTE
225 REM PERCHE' RIPORTA LA FINESTRA A TUTTO SCHERMO !
230 REM PRINT CHR$(27) EQUIVALE A PREMERE IL TASTO ESCAPE
240 :
250 REM PER CREARE UNA FINESTRA FARE GOSUB 400
260 REM FISSATO F ( TRA 1 E 10 ), PER SCRIVERE NELLA FINESTRA F FARE GOSUB 350
270 REM***********************
280 F=0:RI=0:RF=24:CI=0:CF=39:GOSUB500:GOSUB350:REM VEDI COMMENTI
290 REM***********************
300 REM   DEMO : CREA 10 FINESTRE DA UNA COLONNA L'UNA
310 FORF=1TO10:RI=2:CI=3*F+2:RF=24:CF=CI:GOSUB500:NEXT
315 PRINTCHR$(19)CHR$(19)CHR$(147):INPUT"IL TUO NOME";A$
320 PRINTCHR$(19)CHR$(18)"PREMI STOP PER SMETTERE":TRAP520:DO:FORF=1TO10
330 GOSUB350:PRINT"*CIAO "A$SPC(7*RND(1));:NEXT:LOOP
340 :
350 SYS55369:R(AT)=PEEK(2035):C(AT)=PEEK(2036):AT=F
360 POKE2035,RF(F):POKE2036,CF(F):SYS55355:PRINTCHR$(27)"B"CHR$(19);:REM ESC B
370 R=RI(F):DOWHILER:R=R-1:PRINT:LOOP:PRINTSPC(CI(F))CHR$(27)"T";:REM ESC T
380 R=R(F)-RI(F):DOWHILER:R=R-1:PRINT:LOOP:PRINTSPC(C(F)-CI(F));:REM CURSORE
390 RETURN
400 F=0:GOSUB350:PRINT:INPUT"FINESTRA (1-10)";F
410 IFF<=0ORF>10THEN520:REM SERVE F TRA 1 E 10
420 INPUT"RIGA INIZIALE (0-24)";RI
430 IFRI<0ORRI>24THEN520:REM SERVE RI TRA 0 E 24
440 PRINT"RIGA FINALE ("RI;:INPUT"-24)";RF
450 IFRF<RIORRF>24THEN520:REM SERVE RF TRA RI E 24
460 INPUT"COLONNA INIZIALE (0-39)";CI
470 IFCI<0ORCI>39THEN520:REM SERVE CI TRA 0 E 39
480 PRINT"COLONNA FINALE ("CI;:INPUT"-39";CF
490 IFCF<CIORCF>39THEN520:REM SERVE CF TRA CI E 24
500 RI(F)=RI:RF(F)=RF:CI(F)=CI:CF(F)=CF
510 R(F)=RI:C(F)=CI:RETURN:REM CURSORE INIZIALMENTE IN ALTO
520 CLR:PRINTCHR$(19)CHR$(19)CHR$(147):END:REM TRAP
```

```
100 REM NEW MONITOR
110 REM PER C-16
120 :
130 :REM DI C. BAIOCCHI
140 :
150 REM OCCUPA LE POSIZIONI $0333-$0384
160 FORX=DEC("333")TODEC("384"):READA:POKEX,A:U=U+A:NEXT
170 IFU<>11381THENPRINT"ERRORE DI TRASCRIZIONE DATA: CORREGGI":END
180 DATA32,79,255,141,145,27,81,0,32,82,247,165,246,56,32,151,251,76,149,244
190 DATA165,161,24,229,246,141,241,3,165,162,233,,141,242,3,162,3,142,246,3,32
200 DATA79,255,19,27,81,,173,241,3,56,237,246,3,133,161,133,241,173,242,3,233,
210 DATA133,162,133,242,32,82,247,174,246,3,202,240,191,228,246,240,187,208,211
220 KEY1,CHR$(27)+"PG333"+CHR$(27)+"Q"+CHR$(13)
230 PRINT"TENENDO PREMUTO F1 SI HA DISASSEMBLAGGIO NON STOP IN AVANTI"
240 KEY4,CHR$(19)+CHR$(27)+"IG347"+CHR$(27)+"Q"+CHR$(13)
250 PRINT"TENENDO PREMUTO F4 SI HA DISASSEMBLAGGIO NON STOP ALL'INDIETRO"
260 A$=CHR$(27)+"Q"+CHR$(141):FORX=1TO25:A$=A$+CHR$(27)+"I":NEXT
270 A$=A$+CHR$(145)+CHR$(27)+"K"+CHR$(29):KEY8,A$
280 PRINT"CON F8 SI PULISCE LO SCHERMO DALLA POS.DEL CURSORE IN POI"
290 PRINT"IL PROGRAMMA USA LE POSIZIONI $0333-0384"
300 PRINT:PRINT"PUOI FARE NEW E MONITOR":END
```

Smascheriamo il drive

Alcuni trucchi e programmi di utilità indispensabili non solo per conoscere meglio la vostra periferica, ma anche per studiare nuovi tipi di protezioni.

di Giovanni Verrelli

Questo articolo può esser letto anche da chi conosce in modo superficiale il drive 1541. E' doveroso, ad ogni buon conto, precisare che sul numero 22 di C.C.C. abbiamo cominciato il nostro viaggio "nel" dischetto flessibile esaminando le caratteristiche hardware della periferica in oggetto, la funzione della B.A.M. e il modo in cui vengono gestite e memorizzate le varie informazioni su disco.
In effetti si trattava di semplificare alcuni concetti espressi, non proprio chiaramente, nel libretto di istruzioni allegato al drive.

Fu esaminato, in modo approfondito, anche il programma "Display T. & S." che, fornito (su disco) insieme col drive 1541, consente di visualizzare il contenuto di tracce e settori di un qualunque dischetto.

Questo programma, a proposito, sarà necessario utilizzarlo per sfruttare completamente le possibilità di alcuni listati pubblicati in queste pagine.

Inizia lo spettacolo

Con questo articolo, dunque, concluderemo il discorso iniziato, esaminando alcuni comandi che il drive può eseguire unitamente ad alcuni programmi di utilità e ad una piccola raffica di "trucchetti" che permetteranno di mettere in pratica le nozioni apprese.

I più smemorati potranno dare una ripassatina alle pagine 55 e seguenti del numero di luglio/agosto '85 di Commodore Computer Club.

Gli altri, invece, potranno seguirci egualmente a patto di impegnarsi nella lettura. Detto questo, passiamo subito al vivo del discorso.

I comandi del 1541

Il disk drive, oltre ad accettare i comandi SAVE e LOAD, che consentono di salvare e caricare qualsiasi programma, permette l'uso di altri comandi con i quali compiere varie altre operazioni sul dischetto.

Questi comandi, in tutto sei, sono:

NEW: formattazione del dischetto;
INITIALIZE: inizializzazione dischetto;
SCRATCH: cancellazione file
RENAME: cambio nome file
COPY: copia di un file
VALIDATE: sistemazione dischetto.

Vediamo ora come usarli.
Per prima cosa bisogna aprire un canale di comunicazione col drive mediante l'istruzione:

OPEN n,p,15

in cui:
"n" è il numero logico del file che si vuole usare per comunicare (numero variabile, per semplicità, tra 1 e 127 utilizzato successivamente per tutte le operazioni di lettura e scrittura sul canale);
"p" è il numero della periferica (nel caso di un'unica unità collegata p=8);
15 è l'indirizzo secondario che comunica al drive che ci apprestiamo ad inviare comandi.

Da ricordare che per la periferica in esame sono disponibili, come indirizzo secondario, i numeri da 0 a 15. Queste le differenze:

Il numero 0 è usato dall'istruzione LOAD (l'impiego avviene automaticamente ogni qual volta si impartisce il comando citato);
il numero 1 è usato dall'istruzione SAVE e anche in questo caso il suo impiego è "trasparente" in quanto effettuato automaticamente dal sistema ad ogni ordine di SAVE impartito;
i numeri dal 2 al 14 vengono impiegati per le operazioni relative ai file di dati (vedremo più avanti qualche possibile applicazione);
il numero 15, come già detto, è riservato al Sistema Operativo per l'invio dei comandi.
Perciò, l'istruzione:
OPEN 1(o altro numero),8,15
è la sequenza indispensabile da inviare qualora si sia in presenza di una sola unità a dischi collegata al proprio computer.

Fatto questo, il prossimo passo da compiere è quello di inviare sul canale appena aperto una stringa contenente l'iniziale del comando desiderato seguito dagli eventuali altri parametri richiesti.

Per inviare questa stringa si può far uso dell'istruzione PRINT # n, dove n è lo stesso numero logico del file usato nella OPEN, oppure inserire tale stringa subito dopo l'istruzione di apertura del canale.

Si ricorda che l'istruzione PRINT, nella forma PRINT #, non può assolutamente essere abbreviata col punto di domanda (? #), pena l'emissione di un Syntax Error.

Esemplificando il tutto, possiamo eseguire:

OPEN 1,8,15,stringa—comando

oppure

PRINT # 1,stringa—comando

sempre ammesso che 1 sia il numero logico del file usato nella OPEN.

Naturalmente, nel momento in cui avremo terminato di impartire comandi al drive, dovremo chiudere il canale di comunicazione precedentemente aperto usando l'istruzione:
CLOSE n
dove n è sempre il numero logico del file usato nella OPEN (CLOSE1, perciò, nel nostro esempio).
Vediamo ora, uno per uno, i comandi elencati precedentemente.

NEW

Il comando NEW (da non confondere con l'omonima istruzione Basic) è usato per formattare un dischetto, cioè per renderlo "pronto" per ricevere informazioni. Infatti, come certamente già saprete, subito dopo l'acquisto, qualsiasi disco non è ancora idoneo a memorizzare dati, fino a quando non viene praticata l'operazione della formattazione, mediante la quale si ha la creazione, sul dischetto stesso della B.A.M, della directory e di altre informazioni utilizzate dal drive per la corretta gestione del tutto.

I parametri da "passare" al drive sono, oltre all'ovvio comando di formattazione, il nome del disco (cioè una stringa lunga al massimo 16 caratteri), ed i due caratteri dell'ID, che rappresentano una sorta di codice di identificazione del disco molto importante per il DOS (Disk Operative System). Questo, come già detto la volta scorsa, deve esser diverso da dischetto a dischetto in modo da evitare confusioni in fase di "colloquio".

Una buona pratica è quella di cominciare la carriera di disk drivers usando i caratteri 00, 01, 02 e così via fino al 99, continuando (se avete la necessità di utilizzare più di 100 dischi...) coi caratteri A0, A1, ... AA fino a ZZ.

In conclusione, per formattare un disco nuovo, le sequenze possibii sono:

OPEN1,8,15 (nel caso non fosse già stato fatto prima)
seguito da:
PRINT # 1,"N0:nome disco,id"
oppure:
OPEN1,8,15,"N0:nome disco,id"

Da notare che il carattere zero (0) che segue la N (iniziale di NEW) è usato per specificare su quale alloggiamento del drive dovrà essere effettuata l'operazione. Forse non tutti sanno, infatti, che il Commodore 64 e il Vic—20 possono esser collegati, grazie ad una interfaccia, ad un drive doppio Commodore (modelli 4040, 8080 ed altri) i cui due alloggiamenti sono individuati, rispettivamente, dai numeri "0" ed "1".

Nel caso del 1541, ovviamente, questo valore è sempre 0 e può essere omessso.

Per i dischetti già formattati l'uso di questo comando, privo della specificazione dell'ID, sortisce l'effetto di cancellare tutto il contenuto della B.A.M. e della directory, cambiare il nome del disco e lasciare inalterata la ID iniziale col vantaggio di fare tutte queste operazioni impiegando pochi secondi: la formattazione "normale", invece, impiega circa un minuto e mezzo.

Da quanto detto risulta chiaro che qualora si desideri ripulire completamente il dischetto di tutto il suo contenuto impiegando un lasso di tempo molto breve, basterà effettuare:
OPEN1,8,15,"N0:nome disco":CLOSE1
e avremo l'effetto desiderato.

INITIALIZE

Il comando Initialize serve per caricare nella RAM dell'unità a dischi tutte le informazioni necessarie per la corretta gestione del dischetto.

Il suo uso si rivela particolarmente utile qualora, avendo commesso errori vari, il drive sembra non voler più accettare alcun comando. L'inizializzazione del disco (completamente diversa, per carità, dalla formattazione), riporta il drive nelle stesse condizioni in cui si trovava al momento dell'accensione.

Le sequenze possibili sono:
OPEN1,8,15
PRINT #1,"I0"
oppure:
OPEN1,8,15,"I0"

Anche in questo caso lo zero (0) indicante il disco su cui lavorare può essere omesso (il 1541, infatti, lavora con un dischetto solo).

SCRATCH

Il comando Scratch permette di cancellare qualsiasi tipo di file dal dischetto e di rendere disponibili per le memorizzazioni successive tutti i blocchi che esso precedentemente utilizzava.

Sequenze possibili:

OPEN1,8,15
PRINT # 1,"S0:nomefile"
oppure

OPEN1,8,15,"S0:nomefile"

Da notare che è possibile cancellare più file con lo stesso comando ponendo semplicemente, dopo il carattere "S" di Scratch, i nomi dei file non desiderati (quattro, al massimo) separati tra di loro da una virgola.

Perciò, la sequenza:

OPEN1,8,15,"S0:file1,file2,
file3,file4":CLOSE1

cancellerà dal dischetto, con un unico comando, i file dal nome file1, file2, file3 e file4.

Inoltre possono essere utilizzati i due caratteri semplificativi:

Asterisco (*), mediante il quale si rimpiazza la parte finale di un nome, non importa di quanti caratteri;

Punto di domanda (?), per sostituire qualsiasi carattere all'inizio, all'interno o alla fine di un nome.
Perciò, con:

PRINT # 1,"S0:PRO*"

si cancelleranno tutti i file che hanno il nome che comincia con PRO, mentre:

PRINT # 1,"S0:PI??O"

cancellerà tutti i file dal nome di 5 caratteri che inizia con PI e termina con O (PIPPO, PIRRO, PIANO, ecc.)

L'ultima osservazione è che qualora nella directory del disco non sia presente il file specificato con lo SCRATCH, l'operazione sarà eseguita ugualmente (logicamente non verrà cancellato alcun file) ma non ci sarà alcuna segnalazione di errore da parte del drive. Vedremo tra poco cosa questo significa precisamente.

RENAME

Questo comando permette di cambiare il nome ad un file presente su disco, senza cambiarne il posto nella directory.

Subito dopo la R di Rename bisogna inserire il nuovo nome che si vuole assegnare al file (nome che logicamente non deve essere già presente nella directory) seguito dal segno

uguale (=) e dal nome del file da sostituire (nome che, ovviamente, deve essere presente in directory).

Esemplificando, potremo avere:

```
OPEN1,8,15
PRINT #1,"R0:nuovonome=
vecchionome"
```

oppure

```
OPEN1,8,15,"R0:nuovonome=
vecchionome"
```

Da notare che qualora insorga una condizione di errore (nuovonome già presente nella directory, vecchionome non presente in essa oppure nomi con più di 16 caratteri) l'operazione sarà annullata con la conseguente segnalazione di errore da parte del drive (led lampaggiante).

COPY

Il comando Copy serve per copiare un file assegnandogli un nome diverso da quello attuale.

Dato che il 1541, al contrario del 4040, dispone di un solo alloggiamento per dischetti, con questo comando è possibile solamente avere sullo stesso dischetto un file memorizzato due volte, ma con nomi differenti.
Le sequenze possibili sono:

```
OPEN1,8,15
PRINT # 1,"C0:nuovofile=
vecchiofile"
```

oppure

```
OPEN1,8,15,"C0:nuovofile=
vecchiofile"
```

dove nuovofile rappresenta il nome del file—copia che si vuole avere, mentre vecchiofile è il nome del file che si vuole duplicare.

Col comando Copy è possibile anche unire più file sequenziali (quattro al massimo) in un unico nuovo file, somma dei precedenti.

In questo caso la sintassi è la seguente:

```
PRINT# 1,"C0:nuovo=vecchio1,
vecchio2,vecchio3"
```

dove nuovo è il nome del file "destinazione" e vecchio1, vecchio2 e vecchio3 sono i nomi dei file già presenti sul dischetto. Questo comando, pur non segnalando errori, non può essere applicato per la "fusione" di programmi Basic.

Anche in questa occasione qualsiasi errore con i nomi utilizzati annullerà l'operazione in corso e genererà l'appropriata segnalazione di errore da parte del drive.

VALIDATE

Il comando Validate serve per mettere ordine nel disco e far pulizia di tutte le cose "inutili".

Con questo comando è possibile cancellare file aperti ma non richiusi, come i file contraddistinti dalla presenza, nella directory, di un asterisco a fianco della specifica del loro tipo. Riesce quindi a rendere disponibili tutti i blocchi che non risultano occupati nella directory e fa in modo che la somma dei settori liberi e occupati sia 664.
La sintassi è:

```
OPEN1,8,15
PRINT# 1,"V0"
```

oppure

```
OPEN1,8,15,"V0"
```

Da ricordare che questo comando non deve essere utilizzato con dischetti contenenti file random di dati (consultare il manuale d'uso del drive o qualche articolo in materia per saperne di più circa i file random) in quanto si rischia di distruggerli.
U+

Con questo semplice comando è possibile spegnere il led del drive che abbia iniziato a lempeggiare in seguito ad un errore. In questi casi provate a digitare:

```
OPEN 1,8,15,"U+"
CLOSE 1
```

I messaggi di errore

Terminato l'esame dei comandi che il disk drive può accettare, vediamo ora come leggere i messaggi che lo stesso ci invia: è una periferica intelligente, non dimentichiamolo!

La prima cosa da fare è sempre quella di aprire un canale di comunicazione con esso mediante l'ormai nota sequenza:
OPEN1,8,15

Per interrogare la periferica in esame, dobbiamo usare l'istruzione INPUT # che, come dovrebbe essere già noto, lavora soltanto se inserita in un programma Basic. Scriviamo, dunque, le righe Basic necessarie:

```
20 INPUT # 1,CO,TE$,TR,SE
30 PRINTCO;ME$;TR;SE
```

Ogni segnalazione di errore è costituita da un insieme di quattro variabili:
- Un codice di errore (rintracciabile in una delle tabelle del manuale del drive).
- Un messaggio vero e proprio (corrispondente, in breve, al codice precedente).
- Eventuali numeri di traccia e settore in cui si è verificato l'inconveniente (quasi tutti i messaggi, infatti, riguardano errori di traccia e settore).

Per questo motivo con la linea 20 si fa in modo che alla variabile CO sia assegnato il numero del messaggio (il suo codice), alla variabile ME$ la sua descrizione, pur se succinta, e alle variabili TR ed SE, rispettivamente, gli eventuali numeri di traccia e settore dove si è verificato l'errore.

Con la linea 30, invece, si visualizzano le stesse sullo schermo.

Una volta noto il messaggio del drive, per la sua completa interpretazione bisogna fare riferimento alla guida d'uso del 1541, dove è riportato l'elenco di ogni possibile messaggio unitamente al rimedio suggerito.

La lettura del canale degli errori è una pratica che bisognerebbe sempre seguire nella stesura di tutti quei programmi che facciano frequente riferimento al disk drive.

Tracce e settori "in diretta"

Passiamo ora all'esame di altri comandi coi quali è possibile dialogare direttamente coi singoli byte registrati in qualsiasi traccia di qualsiasi settore.

Per l'uso di questi nuovi comandi, oltre all'apertura del canale 15 (la famosa OPEN1,8,15 che abbiamo visto finora) occorre aprire anche un altro canale con indirizzo secondario variabile tra 2 e 14, grazie al quale il DOS mette a disposizione un buffer (:memoria di transito temporanea) da usare per tutte le operazioni di lettura e scrittura.

Spieghiamoci meglio descrivendo, dapprima, problemi poco noti.

L'unità centrale effettua le proprie operazioni ad una determinata velocità dipendente dal circuito elettronico inserito al momento della sua costruzione e da altre caratteristiche fisiche che, in quanto non strettamente ine-

renti l'argomento, non trattiamo in questa sede.

Anche il disk drive dispone di una propria velocità di elaborazione che, inevitabilmente, differisce da quella dell'unità centrale a causa delle diverse caratteristiche fisiche che contraddistinguono due componenti elettronici.

Durante la trasmissione dei dati è necessario che l'unità trasmittente abbia la stessa velocità di quella ricevente (pena l'inevitabile perdita di dati). Poichè il computer e il disk drive, purtroppo, non possiedono egual velocita', il problema viene risolto utilizzando un buffer, vale a dire una memoria temporanea di transito, che accoglie i dati trasmessi e li conserva fino a quando non vengono prelevati dall'unità ricevente.

Questo è il motivo per cui, per poter colloquiare direttamente con i singoli byte registrati su disco, si ha bisogno di un buffer.
Per raggiungere i nostri scopi, la tecnica da usare è la seguente:

1 — chiedere al DOS di mettere a disposizione un buffer per le nostre operazioni;
2 — riportare in tale buffer il settore contenente il byte (oppure i byte) da modificare;
3 — modificare il byte (o i byte);
4 — riscrivere su disco il settore così modificato.

Ritornando al discorso iniziale, la prima operazione da compiere è l'apertura del canale comandi con:

OPEN1,8,15

In seguito, con l'istruzione:

OPEN2,8,2,"#"

si comunica al DOS l'intenzione di voler colloquiare coi singoli byte e di riservare, perciò, un buffer per le nostre operazioni.

Il valore 2 posto subito dopo la OPEN rappresenta il numero logico del file. Anche in questo caso esso può variare tra 1 e 127 e logicamente deve essere differente da quello utilizzato nell'apertura del canale comandi (pena un File Open Error).

Il numero 8 che segue rappresenta il numero di periferica, e l'altro numero (anch'esso 2), l'indirizzo secondario variabile, comunque, a piacimento tra 2 e 14. La stringa speciale del cancelletto (#) è utilizzata per richiedere il buffer.

Effettuata l'apertura dei due canali, presupposto indispensabile per tutte le successive operazioni, vediamo ora quali sono e come si usano i comandi che permettono questo "accesso diretto" ai byte del disco.

U1 (oppure UA)

Il comando U1 (o UA) serve per trasferire nel buffer riservatoci dal DOS i 256 byte costituenti l'intero settore specificato con i parametri che seguono tale comando.

La sua corretta sintassi è:

PRINT # n,"U1:"ca;dr;tr;se

in cui:

n è il numero logico del canale comandi precedentemente aperto (1 nel nostro caso);
ca è il numero logico del file aperto per il buffer (2 nel nostro caso);
dr è il numero del drive (sempre 0 per il 1541);
tr è il numero di traccia del blocco che vogliamo trasferire nel buffer;
se è, infine, il numero di settore del blocco che vogliamo trasferire nel buffer.

Volendo, ad esempio, trasferire nel buffer il settore 0 della traccia 18, dovremo effettuare (sempre dopo le 2 OPEN iniziali):

PRINT # 1,"U1:"2;0;18;0

Da notare che se i parametri da passare insieme al comando sono delle costanti, possono essere inseriti all'interno delle virgolette (") separati tra di loro dalla virgola (,), mentre se tali parametri sono delle variabili, allora è necessario che siano inseriti all'esterno delle virgolette separati tra di loro dal punto e virgola (;).

U2 (oppure UB)

Il comando U2 (o UB) serve per trasferire il contenuto del buffer nel settore specificato dai parametri che lo seguono (si tratta, in pratica, del comando opposto ad U1).
La sua sintassi è:

PRINT # n,"U2:"ca;dr;tr;se

con le identiche precisazioni del comando U1.

B—P

Il comando B—P (Buffer—Pointer) permette di accedere direttamente su qualsiasi carattere del buffer sia per operazioni di scrittura che di lettura.

Con il comando B—P, infatti, si assegna un valore al puntatore del buffer. Tale valore rappresenta un contatore che indica, in ogni momento, la posizione, all'interno del buffer, del prossimo byte da leggere (oppure scrivere), dando così la possibilità di un libero accesso a qualsiasi carattere.

Dato che un buffer contiene un blocco, e che un blocco è costituito da 256 byte numerati da 0 a 255, appare subito evidente che il valore che si può assegnare a tale puntatore può variare tra 0 (primo carattere del buffer) e 255 (ultimo carattere del buffer).

Il formato del comando B—P è:

PRINT # n,"B—P:"ca;po

dove:

n è il numero logico del canale comandi (sempre 1 nei nostri esempi);
ca è il numero del canale associato al buffer (2);
po è la posizione che si vuole assegnare al puntatore del buffer (come già visto, può assumere valori da 0 a 255).

Dopo aver posizionato il puntatore, per trasferire le informazioni da buffer a memoria centrale si può far uso dell'istruzione GET # sul canale associato al buffer (GET # 2 per i nostri esempi), mentre per scrivere informazioni nel buffer bisogna far uso dell'istruzione PRINT # sempre sul canale associato al buffer (PRINT # 2).

B—A

Questo comando, abbreviazione di Block—Allocate, fa in modo che il settore da esso specificato risulti "allocato", risulti cioè già occupato e perciò non libero per le future esigenze del sistema.
La sua sintassi è:

PRINT #n,"B—A:"dr;tr;se

dove:

n è il numero del canale comandi;
dr è il numero del drive;
tr è la traccia del blocco da allocare;
se è il settore del blocco da allocare.

Se il blocco specificato è già occupato, allo-

ra il sistema produce l'errore 65 (No Block) e fornisce nelle variabili TR e SE (le ultime due lette sul canale errori viste in precedenza) l'indirizzo del primo blocco disponibile (TR=0 se il dischetto è pieno).

B—F

Il comando B—F (Block—Free) effettua l'operazione opposta a B—A: libera un settore già occupato e lo rende disponibile al sistema.

Il suo formato è:

PRINT # n,"B—F:"dr;TR;se;

con gli stessi simboli di B—A.

Qualora il blocco sia già libero il sistema non produce alcun errore in quanto nell'esecuzione di questo comando ogni blocco si intende occupato.

Da tenere ben presente che anche dopo l'uso di questi comandi (come del resto sempre dopo aver aperto dei files) è indispensabile chiudere i canali aperti usando le istruzioni:
CLOSE2
CLOSE1
dove 2 e 1 sono i numeri usati nell'apertura del canale—buffer e del canale comandi.

Dalla teoria alla pratica

Terminata la parte "teorica" dell'articolo, vediamo ora alcune applicazioni pratiche nelle quali useremo i comandi esaminati in precedenza.

Esaminiamo il modo in cui è possibile cambiare il nome del disco senza cancellarne il contenuto.

Normalmente l'unica operazione che rende possibile variare il nome del disco è la riformattazione, la quale, però, provoca la cancellazione di tutto ciò che il dischetto contiene.

Per superare l'ostacolo possiamo comunque agire direttamente sulla traccia 18 settore 0 sostituendo il nome del disco, ivi memorizzato, con una nuova stringa.

Il programma che effettua tale modifica è il N.1 che commentiamo qui di seguito:

Con le linee 50—70 si fa in modo che la variabile N$ contenga il nuovo nome del disco completato (se di lunghezza minore di 16 caratteri) dal necessario numero di spazi shiftati (codice ASCII: 160).

80—90: apertura del canale comandi (1) e del canale associato al buffer (2).

Dalla linea 100 comincia il "corpo" principale del programma:

- Si trasferisce la traccia 18 settore 0 nel buffer (linea 100);
- Si posiziona il puntatore sul 144esimo carattere, in cui, lo abbiamo visto la volta scorsa, risiede il nome del disco (linea 110);
- Si scrivono, a partire da questa posizione, i 16 caratteri del nuovo nome del disco (linea 120);
- Si riporta su disco il contenuto del buffer così modificato nella traccia 18 settore 0 (linea 130);
- Si inizializza il dischetto (linea 140);
- Si chiudono i 2 canali aperti all'inizio (linee 150—160).

Da notare che alla linea 120 l'ultimo carattere è un punto e virgola (;) per evitare che il sistema, dopo aver scritto la variabile N$, inserisca automaticamente un carattere di ritorno carrello CHR$(13). Dovrebbe essere noto, infatti, che se dopo un PRINT o PRINT # non c'è alcun carattere separatore, il sistema pone automaticamente, anche se non richiesto, un CHR$(13) con tutte le conseguenze che può comportare.

Il comando di linea 140 (inizializzazione) assicura, inoltre, il pieno successo dell'operazione.

Usando questo stesso metodo alcuni programmi mal strutturati hanno la pretesa di cambiare anche i due caratteri dell'ID del disco limitandosi a variare i bytes 162—163 della traccia 18 settore 0.

Inutile dire che così facendo si cambia solo l'ID che compare nella directory senza per questo intervenire sulla "reale" ID del disco, scritta su ogni traccia e settore durante la formattazione: solamente con la riformattazione del dischetto è possibile cambiare ID al disco.

L'uso di questi programmi, inoltre, è controproducente in quanto fa credere che il disco abbia una determinata ID che invece differisce da quella reale, rendendo possibile così l'uso di due dischetti che, anche se a prima vista hanno ID differenti, in realtà possiedono ID uguali.

Il programma 2 è di ausilio a tutti coloro che hanno usato questi programmi per la variazione dell'ID del disco.

Con esso si legge la reale ID del disco da un blocco qualsiasi (si è scelta la traccia 18 settore 0) e la si riscrive correttamente nella directory.

La tecnica utilizzata per la lettura dell'ID da un blocco qualsiasi fa uso di alcuni comandi che permettono l'accesso diretto alla RAM del 1541. Data l'estrema complessità dell'argomento, si è preferito non trattare tali comandi nell'articolo.

Il terzo programma dispone di quattro opzioni:
1 — Recupero di un file cancellati accidentalmente.
2 — Cancellazione diretta di un file.
3 — Protezione di un file da indesiderati comandi di cancellazione (scratch).
4 — Lettura dei settori utilizzati in un file.

Esaminiamole brevemente una per una.

L'opzione 1 permette il recupero di un file involontariamente cancellato.

Quando il DOS riceve l'ordine di cancellare un file, pone a 0 il byte indicante il suo tipo (PRG, SEQ, USR o REL) e rende disponibili, per successive memorizzazioni, i blocchi da esso utilizzati fino a quel momento.

Il programma non fa altro che rimettere, nel byte indicante il tipo di file, il valore originario (129 per SEQ, 130 per PRG, 131 per USR e 132 per REL). Oltre a questa operazione il programma provvede a rendere nuovamente occupati, nella B.A.M., i blocchi utilizzati per la sua memorizzazione.

Il successo dell'operazione è sicuro se il programma viene usato subito dopo lo Scratch errato, mentre le possibilità di recupero diminuiscono a mano a mano che i file memorizzati dopo la cancellazione aumentano. I blocchi resi disponibili con l'operazione di cancellazione, potrebbero, infatti, essere stati utilizzati dai files memorizzati successivamente; oppure i 30 bytes contenenti tutte le informazioni del file cancellato potrebbero essere stati utilizzati per la definizione di un file successivo.

L'opzione 2 (riga 80) permette la cancellazione diretta del file (pone a 0 il byte indicante il suo tipo) ed è utile quando, anche usando il comando Scratch, non si riesce in alcun modo a cancellare un file presente su disco. Benchè ciò possa sembrare strano, questa eventualità si verifica quando il nome del file è costituito da uno o più caratteri "proibiti", come ad esempio la virgola (,), l'asterisco, il punto interrogativo.

Con la terza opzione (riga 90) si protegge qualsiasi file dallo Scratch accidentale. Esso, sommando 64 al valore del byte indicante il tipo di file, fa in modo che lo Scratch praticato su questo file "protetto" non sortisca alcun effetto. Per cancellare un file protetto in questo modo si può solo far ricorso all'opzione 2 del programma.

L'ultima opzione permette la stampa su video di traccia e settore dei blocchi utilizzati nella memorizzazione del file.

La tecnica utilizzata è di cominciare con la lettura del primo blocco del file (puntato dai bytes 1 e 2 di ogni "entrata") e proseguire leggendo il puntatore al prossimo blocco (primi due bytes di ogni settore) fino a quando non si incontra il blocco con traccia 0, che rappresenta l'ultimo blocco del file.

Utilizzando la stessa tecnica sono mostrati anche i blocchi contenenti i Side Sectors utilizzati nei file relativi.

Da ricordare che con la pressione del tasto funzione F7 si sospende la visualizzazione fino alla pressione di un tasto qualsiasi, mentre premendo F1 si ritorna al menù principale.

L'utilizzazione del programma richiede la conoscenza del settore della traccia 18 dove sono contenuti i 30 bytes di definizione del file (i bytes, per intenderci, indicanti nome del file, il tipo, il numero di blocchi utilizzati, eccetera) e il numero d'ordine, tra le possibili 8 posizioni, di tale "entrata" nel settore.

Per ottenere queste due informazioni si può far uso, prima di usare questo programma, del DISPLAY T&S (cui accennavamo all'inizio dell'articolo) cominciando con la lettura della traccia 18 settore 1 e proseguendo, se necessario, nella lettura dei blocchi successivi fino a trovare l'"entrata" che interessa.

Il programma 4 permette di allocare (=occupare) o deallocare (=render libero) qualsiasi blocco con l'unica esclusione di quelli della traccia 18. E' sempre meglio, infatti, evitare la manipolazione di tali settori.

Il programma, dopo aver chiesto se si desidera allocare o deallocare il blocco, ne richiede l'indirizzo di traccia e settore ed effettua l'operazione desiderata.

Da ricordare che, qualora nell'allocazione del settore il sistema produce l'errore 65 (No Block), gli ultimi due valori stampati su video corrispondono alla traccia e al settore del primo blocco disponibile.

Il programma può essere usato anche per verificare se nella B.A.M. un determinato settore è libero o già occupato. Per far ciò si cerca di allocare tale blocco: se vien fuori l'errore 65 si deduce che è occupato, altrimenti è libero.

Data l'estrema semplicità del listato, non si riporta alcuna spiegazione.

Il programma 5, che conclude questa serie di listati di utilità, permette di conoscere l'indirizzo iniziale e finale di qualsiasi programma presente su disco.

Viene aperto il file programma come se fosse un file di dati (è usato come indirizzo secondario il numero 3): i primi due bytes letti corrispondono al byte basso ed al byte alto dell'indirizzo iniziale. Tale valore è calcolato applicando l'ormai nota formula: byte basso + 256 * byte alto.

Per conoscere l'indirizzo finale si continua a leggere, byte per byte, il contenuto del programma e si incrementa, nello stesso tempo, l'indirizzo iniziale fino a quando non si incontra il segnale di fine file (ST=64), nel qual caso si stampa il valore così ottenuto e si terminano le operazioni.

Da ricordare che la variabile di stato ST, ampiamente trattata in un precedente articolo (pagina 55 e seguenti di C.C.C. N. 21), è usata nei sistemi Commodore per restituire informazioni sull'esito dell'ultima operazione di I/O eseguita. Quando essa assume il valore 64 vuol dire che è stato letto l'ultimo carattere di un file e che perciò non vi sono più altri dati da leggere.

Tornando al listato N.5, al RUN verrà richiesto il nome del programma presente su disco di cui si vuol conoscere la locazione iniziale e finale. In seguito il valore iniziale sarà subito stampato su video seguito da un valore che si incrementerà fino a quando non sarà raggiunta la fine del programma.

Dato che per programmi occupanti più blocchi il tempo impiegato per ricavare il valore finale è rilevante, si consiglia la compilazione del programma (ottimo per questo scopo il Petspeed) in modo da diminuire i tempi di attesa.

Importante: i programmi pubblicati manipolano direttamente i bytes registrati su disco. Prima di mandarli in esecuzione è bene assicurarsi che non contengano errori nè omissioni poichè questi potrebbero essere fatali al contenuto del dischetto. Esercitatevi con un dischetto nuovo che, formattato e "riempito" di dati e programmi vari, servirà da cavia per i vostri esperimenti.

Un paio di trucchi

Come promesso all'inizio, concludiamo l'articolo presentando alcuni "trucchetti" che il drive 1541 permette di effettuare.

Il primo di questi rende la directory di qualsiasi disco non più leggibile, rivelandosi perciò particolarmente utile come (modesta) protezione per i propri dischetti. Non potendo leggere il nome di un programma non si può nemmeno caricarlo!

Per fare questo basta semplicemente far precedere il nome del primo file che compare nella directory da tre caratteri CHR$(0).

Ipotizzando che il nome del primo file registrato su disco sia PIPPO, la sequenza di comandi:

```
A$=CHR$(0)+CHR$(0)
+CHR$(0)
OPEN1,8,15,"R0:"+A$+"
PIPPO=PIPPO"
CLOSE1
```

sortirà l'effetto desiderato.

Dopo queste operazioni bisogna solamente ricordare che per caricare il programma col nome alterato (PIPPO nel nostro esempio) è necessario effettuare:

```
LOAD"*",8
```

con cui si carica il primo programma presente

su disco, oppure:

LOADCHR$(0)+"*",8

invece del solito:

LOAD"nome programma",8

Il secondo "trucchetto" permette di aggiungere commenti al nome di un programma senza che questi facciano parte del nome stesso.

Spieghiamoci meglio: normalmente, quando si ha necessità di inserire dei commenti al nome di un programma (SYS di partenza, numero di versione, ed altro) si è soliti inserire gli stessi subito dopo il nome del file (es. HESMON 32768, EAT 2051) con lo svantaggio che questi diventano parte integrante del nome del file che bisogna obbligatoriamente digitare nella fase di caricamento, pena la comparsa del messaggio "File not found error" e il lampeggio della spia rossa del drive.

C'è un modo per evitare questo.

In fase di salvataggio del programma su disco, dopo aver specificato il nome del file, inserite uno spazio shiftato (tenendo premuto il tasto SHIFT premere la barra spaziatrice) seguito dal commento e dalla chiusura delle virgolette (ricordate che il nome del file, lo spazio shiftato e il commento non devono superare i 16 caratteri).

Al momento della visualizzazione della directory si vedrà il nome del programma racchiuso tra le virgolette (") e il commento alla sua destra.

Questi programmi possono essere caricati specificando solamente il nome del file: il commento può essere tranquillamente ignorato.

Con questa procedura, inoltre, è possibile anche registrare su disco programmi con nomi uguali ma commenti differenti.

Naturalmente in fase di LOAD sarà caricato solamente il primo dei programmi con nomi uguali, a meno che non si specifichi anche il commento digitando:

LOAD"nome prg(spazio shiftato)commento",8

L'ultimo "trucchetto" permette di recuperare un file appena cancellato su disco. Questo è possibile semplicemente digitando, subito dopo lo Scratch, il comando:
LOAD"*",8
assicurandosi che subito dopo la cancellazione non sia stata effettuata alcuna operazione di accesso al drive o che il computer (o il drive) non siano stati spenti.

Per verificare quanto detto scrivete il programmino:
10 PRINT"TUTTO O.K."
e salvatelo su disco col nome PROVA digitando:
SAVE"PROVA",8

Verificate, caricando e listando la directory, che il file PROVA sia stato salvato su disco. Effettuate poi un NEW per cancellare il contenuto della memoria e digitate:
OPEN1,8,15,"S0:PROVA":CLOSE1
per cancellare dal disco il file PROVA.

Fatto questo, digitando:
LOAD"*",8
ed effettuando un LIST si potrà vedere come effettivamente sia stato caricato il programma appena cancellato che, se necessario, può essere nuovamente salvato su disco col nome che aveva precedentemente o usando un altro nome.

Prima di concludere, diamo una informazione che certamente sarà da molti ben accetta: i file USER, di cui spesso si sente parlare ma di cui non si trova menzione nemmeno nel manuale d'uso del drive, non sono altro che normali file sequenziali con specifica USR invece di SEQ.

Il loro uso (apertura e chiusura file, scrittura e lettura dati) è identico a quello dei file sequenziali con l'unica eccezione dell'indicazione della lettera "U" al posto della "S" nella OPEN iniziale.

Perciò, il programma:

```
10 OPEN2,8,2,"PIPPO,S,W"
20 INPUTB$:IFB$=
   "*"THENCLOSE2:END
30 PRINT # 2,B$:GOTO20
```

scriverà su file sequenziale (SEQ) di nome PIPPO i dati man mano introdotti da tastiera fino a quando non si introdurrà il carattere asterisco (*), mentre con:

```
10 OPEN2,8,2,"PIPPO,U,W"
20 INPUTB$:IFB$=
   "*"THENCLOSE2:END
30 PRINT # 2,B$:GOTO20
```

si produrrà lo stesso effetto con l'unica differenza che i dati saranno scritti su file user (USR).

Anche per la lettura dei dati basterà semplicemente inserire la lettera "S" o la lettera "U" nella OPEN iniziale sempre a seconda che si sia in presenza, rispettivamente, di file sequenziale o file user.

Esemplificando, per i file squenziali avremo:

```
10 OPEN2,8,2,"PIPPO,S,R"
20 INPUT # 2,B$:PRINTB$
30 IFST=64THENCLOSE2:END
40 GOTO20
```

mentre per i file user:

```
10 OPEN2,8,2,"PIPPO,U,R"
20 INPUT #2,B$:PRINTB$
30 IFST=64THENCLOSE2:END
40 GOTO20
```

```
20 REM  PROGRAMMA 1
30 REM  CAMBIO NOME DISCO
40 :
50 FOR I=1 TO 16:A$=A$+CHR$(16
   0):NEXT
60 INPUT "NUOVO NOME DISCO";N$
70 N$=LEFT$(N$+A$,16)
80 OPEN 1,8,15
90 OPEN 2,8,2,"#"
100 PRINT#1,"U1:"2;0;18;0
110 PRINT#1,"B-P:"2;144
120 PRINT#2,N$;
130 PRINT#1,"U2:"2;0;18;0
140 PRINT#1,"I0"
150 CLOSE 2
160 CLOSE 1
```

```
2Ø REM  PROGRAMMA 2
3Ø REM  RISCRITTURA REALE 'ID'
    DISCO
4Ø :
5Ø OPEN 1,8,15
6Ø OPEN 2,8,2,"#"
7Ø PRINT#1,"U1:"2;Ø;18;Ø
8Ø PRINT#1,"M-R";CHR$(22);CHR$
    (Ø)
9Ø GET #1,A$
1ØØ PRINT#1,"M-R";CHR$(23);CHR$
    (Ø)
11Ø GET #1,B$
12Ø PRINT#1,"B-P:"2;162
13Ø PRINT#2,A$;B$;
14Ø PRINT#1,"U2:"2;Ø;18;Ø
15Ø PRINT#1,"IØ"
16Ø CLOSE 2
17Ø CLOSE 1
```

```
 1Ø REM  PROGRAMMA 3
 2Ø REM  MANIPOLAZIONE FILE SU
    DISCO
 25 REM  BY GIOVANNI VERRELLI
 27 REM  GIOIA DEL COLLE (BA)
 3Ø :
 4Ø REM  POKE5328Ø,11:POKE53281
    ,Ø:POKE646,3: PER C-64
 5Ø DIM TE$(2Ø)
 6Ø PRINT"[CLEAR][2 DOWN]" TAB(
    14)"---MENU---":PRINT
 7Ø PRINT" 1: RECUPERO FILE CAN
    CELLATO"
 8Ø PRINT" 2: CANCELLAZIONE DIR
    ETTA FILE"
 9Ø PRINT" 3: PROTEZIONE FILE D
    A SCRATCH"
1ØØ PRINT" 4: LETTURA SETTORI U
    TILIZZATI FILE"
11Ø PRINT" 5: RITORNO AL BASIC"
12Ø PRINT"[DOWN]QUALE SCEGLI ?
    ";
13Ø GET A$:IF A$<"1" OR A$>"5"
    THEN 13Ø
14Ø PRINTA$:PRINT:A=VAL(A$)
15Ø IF A=5 THEN PRINT"ARRIVEDER
    CI !!":END
16Ø SE=Ø:INPUT "SETTORE ";SE
17Ø IF SE<1 OR SE>18 THEN 16Ø
18Ø ND=Ø:INPUT "N. D'ORDINE NEL
     SETTORE (1-8) ";ND
19Ø IF ND<1 OR ND>8 THEN 18Ø
2ØØ OPEN 1,8,15,"IØ":GOSUB 82Ø
21Ø OPEN 2,8,2,"#":GOSUB 82Ø
22Ø ND=ND-1:TR=18:PO=2+ND*32
23Ø GOSUB 74Ø
24Ø ON AGOTO 5ØØ,56Ø,59Ø
25Ø REM  ** SETTORI UTILIZZATI
    FILE **
26Ø FOR I=Ø TO 2Ø
27Ø GET #2,B$:IF B$="" THEN B$=
    CHR$(Ø)
28Ø TE$(I)=B$
29Ø NEXT
3ØØ NF$="":FOR I=3 TO 18:NF$=NF
    $+TE$(I):NEXT
31Ø PRINT"[DOWN]NOME FILE: [RVS
    ]"NF$
32Ø TR=ASC(TE$(1)):SE=ASC(TE$(2
    )):BL=Ø:FL=Ø:PO=Ø
33Ø PRINT"[DOWN][RVS]SETTORI UT
    ILIZZATI:"
34Ø T$=MID$(STR$(TR),2):S$=MID$
    (STR$(SE),2)
35Ø PRINTT$"-"S$,:BL=BL+1
36Ø GET D$:IF D$=CHR$(133) THEN
     67Ø
37Ø IF D$=CHR$(136) THEN GOSUB
    78Ø
38Ø GOSUB 74Ø
39Ø GET #2,TR$,SE$
4ØØ TR=ASC(TR$+CHR$(Ø)):SE=ASC(
    SE$+CHR$(Ø))
41Ø IF TR THEN 34Ø
42Ø IF FL THEN 47Ø
43Ø TR=ASC(TE$(19)):SE=ASC(TE$(
    2Ø)):FL=1
44Ø IF TR=Ø THEN 47Ø
45Ø PRINT:PRINT"[RVS]SIDE SECTO
    RS        :[RVOFF]"
46Ø GOTO 34Ø
47Ø PRINT:PRINT"[RVS]BLOCCHI OC
    CUPATI   :[RVOFF]"BL
48Ø GOTO 67Ø
49Ø REM  ** RECUPERO FILE CANCE
    LLATO **
5ØØ PRINT"[DOWN]1=SEQ 2=PRG 3=U
    SR 4=REL"
51Ø TY=Ø:INPUT "TIPO FILE ";TY
52Ø IF TY<1 OR TY>4 THEN 51Ø
```

```
530 B=TY+128
540 GOTO 620
550 REM  * CANCELLAZIONE DIRETT
    A FILE *
560 B=0
570 GOTO 620
580 REM  * PROTEZIONE FILE DA S
    CRATCH *
590 GET #2,B$:B=ASC(B$+CHR$(0))
600 B=B+64
610 GOSUB 750
620 PRINT#2,CHR$(B);:GOSUB 820
630 PRINT#1,"U2:"2;0;TR;SE:GOSU
    B 820
640 IF A=3 THEN 670
650 PRINT:PRINT"VALIDATE IN COR
    SO ..."
660 PRINT#1,"V0":GOSUB 820
670 PRINT#1,"I0":GOSUB 820
680 CLOSE 2:CLOSE 1
690 IF A<>4 THEN 720
700 PRINT:PRINT"PREMI UN TASTO
    PER CONTINUARE ..."
710 GOSUB 780
720 GOTO 60
730 REM  * LEGGE SETTORE E POSI
    Z.PUNT. *
740 PRINT#1,"U1:"2;0;TR;SE:GOSU
    B 820
750 PRINT#1,"B-P:"2,PO:GOSUB 82
    0
760 RETURN
770 REM  ** ATTENDE PRESSIONE T
    ASTO **
780 POKE 198,0: REM  PER C-64
790 GET C$:IF C$="" THEN 790
800 RETURN
810 REM  ** LETTURA CANALE ERRO
    RI **
820 INPUT#1,CO,ME$,T,S
830 IF CO=0 THEN RETURN
840 PRINT:PRINTC0;ME$;T;S
850 CLOSE 2:CLOSE 1:STOP
```

```
 20 REM  PROGRAMMA 4
 30 REM  ALLOCAZ./DEALLOCAZ. BL
    OCCO
 40 :
 50 A$(1)="B-A:":A$(2)="B-F:"
 60 REM  POKE53280,11:POKE53281
    ,Ø:POKE646,3: PER C-64
 70 PRINT"[CLEAR][2 DOWN]" TAB(
    14)"---MENU---":PRINT
 80 PRINT" 1: ALLOCAZIONE BLOCC
    O"
 90 PRINT" 2: DISALLOCAZIONE BL
    OCCO"
100 PRINT" 3: RITORNO AL BASIC"
110 PRINT"[DOWN]QUALE SCEGLI ?
    ";
120 GET A$:IF A$<"1" OR A$>"3"
    THEN 12Ø
130 PRINTA$:PRINT:A=VAL(A$)
140 IF A=3 THEN PRINT"ARRIVEDER
    CI !!":END
150 TR=Ø:INPUT "TRACCIA ";TR
160 IF TR<1 OR TR>35 OR TR=18 T
    HEN 15Ø
170 SE=Ø:INPUT "SETTORE ";SE
180 IF SE<Ø OR SE>2Ø THEN 17Ø
190 PRINT
200 OPEN 1,8,15,"IØ":GOSUB 3ØØ
210 OPEN 2,8,2,"#":GOSUB 3ØØ
220 PRINT#1,A$(A);Ø;TR;SE:GOSUB
     3ØØ
230 CLOSE 2:CLOSE 1
240 PRINT"OPERAZIONE COMPLETATA
    ."
250 PRINT"UN TASTO PER CONTINUA
    RE ..."
260 POKE 198,Ø:REM  PER C-64
270 GET B$:IF B$="" THEN 27Ø
280 GOTO 7Ø
290 REM  *** LETTURA CANALE ERR
    ORI ***
300 INPUT#1,CO,ME$,T,S
310 IF CO=Ø THEN RETURN
320 PRINTCO;ME$;T;S
330 CLOSE 2:CLOSE 1:STOP
```

```
 20 REM  PROGRAMMA 5
 30 REM  LOCAZ. INIZIO E FINE P
    RG
 40 :
 50 REM  POKE53280,11:POKE53281
    ,Ø:POKE646,3: PER C-64
 60 PRINT"[CLEAR] --INDIRIZZO I
    NIZIO E FINE PROGRAMMA--":P
    RINT
 70 NF$="":INPUT "NOME PROGRAMM
    A ";NF$
 80 IF NF$="" OR LEN(NF$)>16 TH
    EN 7Ø
 90 OPEN 1,8,15,"IØ":GOSUB 25Ø
100 OPEN 3,8,3,NF$+",P":GOSUB 2
    5Ø
110 GET #3,B$,C$
120 LI=ASC(B$+CHR$(Ø))+256*ASC(
    C$+CHR$(Ø))
130 PRINT"[2 DOWN][RVS]LOCAZION
    E INIZIALE:[RVOFF]";LI
140 PRINT"-------------------"
150 LF=LI
160 PRINT"[RVS]LOCAZIONE FINALE
      :[RVOFF]"
170 PRINT"[UP]" TAB(19)LF
180 IF ST=64 THEN 22Ø
190 GET #3,B$
200 LF=LF+1
210 GOTO 17Ø
220 CLOSE 3:CLOSE 1
230 PRINT"[2 DOWN]OPERAZIONE CO
    MPLETATA."
240 END
250 INPUT#1,CO,ME$,TR,SE
260 IF CO=Ø THEN RETURN
270 PRINT"[2 DOWN]"CO,ME$,TR;SE
280 CLOSE 3:CLOSE 1:STOP
```

Reversi: un gioco per pensare

di Flavio Molinari

Sembra che questo gioco abbia avuto origine alla fine del secolo scorso in Inghilterra (in forma un po' diversa da quella attuale); oggi si gioca sulle solite scacchiere standard con 32 pedine per giocatore, colorate in modo diverso sui due lati.

Per chi non le conosce, illustro brevemente le regole:
- ogni giocatore dispone di 32 pezzi, colorati in nero su un lato ed in bianco sull'altro. Si inizia con la scacchiera vuota ed a turno si depongono le pedine. Le prime quattro devono essere disposte nelle quattro caselle centrali (vedi figura);
- si deve cercare di "catturare" i pezzi avversari, facendo in modo che vengano a trovarsi tra due dei nostri. Una volta catturati non vengono tolti, ma rivoltati sull'altro lato: diventano così "amici".

Le catture possono essere fatte in senso orizzontale, verticale o diagonale ed anche, se la mossa è particolarmente abile, in più direzioni contemporaneamente;
- la partita finisce quando tutte le caselle sono occupate o nessuno dei due contendenti può più muovere. Vince chi avrà in campo il maggior numero di pezzi del proprio colore.

Come potete vedere, le regole sono molto semplici. Viceversa il gioco è abbastanza difficile da analizzare e non sempre si può stabilire se una posizione è favorevole ad uno piuttosto che all'altro giocatore. I colpi di scena sono infatti molto frequenti e spesso bisogna attendere l'ultima mossa per essere certi di aver vinto (o perso!).

Questo che vi presentiamo è certamente uno dei giochi da scacchiera dei più moderni ed interessanti. Potete sfidare i vostri amici oppure cimentarvi contro il computer stesso.

Note per copiare il listato

Il programma è interamente in basic e può girare su qualsiasi computer Commodore, che disponga di uno schermo di almeno 40 colonne per 25 righe. Originariamente scritto per il Commodore 64, è possibile adattarlo ad altri computer (C-16 e PLUS-4), con le seguenti, piccole modifiche:
- digitare in modo diretto il comando:
COLOR 0,7,5
- scrivere la riga 1920 come segue:
1920 RETURN

Questa corrisponde all'inizio di una breve routine sonora (un beep), utile ma non indispensabile in alcune fasi del gioco. Così facendo, il programma ignorerà tutto quello che segue, ma è implicito l'invito ai possessori di C16 a modificare le linee 1920 e 1930 ed inserire un effetto sonoro di propria creazione.

Sostituite, a tale scopo, le varie POKE, con le apposite istruzioni presenti nel BASIC 3.5, quali VOL, SOUND...

Infine, il listato è numerato a partire da 100, con incrementi di 10: attivando la funzione AUTO, risparmierete parecchio tempo per la battitura (per il C64 il discorso vale solo se disponete di un'opportuna utility o di un'espansione del tipo BASIC 4.0).

Per iniziare

Date il RUN e rispondete ad un paio di domande:
quanti giocatori ? (1 o 2)

Se decidete di sfidare il computer, questo vi chiederà (bontà sua...), di scegliere il colore delle pedine. Come di consueto i bianchi sono sempre i primi a muovere.

La scacchiera ha il solito formato di 8 per 8 caselle; uniche differenze rispetto a quelle che siamo abituati vedere sono la mancanza della tipica colorazione a quadri bianchi e neri e la numerazione delle case da 1 a 64. Per "posare" una pedina, basterà battere un numero compreso in questo intervallo (più il tasto RETURN).

Attenzione, però se eseguite una mossa che non rispetta le regole sopra descritte, perdete il turno. Una piccola cattiveria per non permettere di barare, ma sopratutto per stimolare i neofiti ad apprendere più velocemente le regole del gioco. Comunque, niente paura: dopo due o tre partite perse miseramente, sa-

prete padroneggiare pefettamente il gioco.

Un'ultima precisazione: a volte, in alcune fasi della partita, è più conveniente passare il turno, anzichè fare una mossa che magari può essere compromettente. In questo caso siamo obbligati a muovere comunque. Per rendere più rapida l'esecuzione del programma, non vengono effettuati controlli in tal senso: perciò cercate di non approfittarne troppo...

Qualche variabile interessante

Per chi vuol saperne di più su come è strutturato il programma, diamo qui di seguito alcune informazioni relative alle variabili adoperate.

D$(): posiziona il cursore. Usata con la funzione TAB(), simula l'istruzione PRINT AT(). Esempio:
PRINT D$(15)TAB(20)

posiziona il cursore alla quindicesima riga e ventesima colonna.

PD$ visualizza pedina su video. Cambiandone opportunamente il colore si potrà posizionare, voltare o cancellare.

A$,B$,C$,D$: stringhe per il disegno della scacchiera.

P1(), P4(). Queste matrici permettono al computer di "vedere" lo stato della scacchiera: P1() conterrà 0 se la casella corrispondente è vuota, oppure 1 o 2 se è occupata dal nero o dal bianco. P4() è utilizzata dal programma per la ricerca di una posizione lecita.

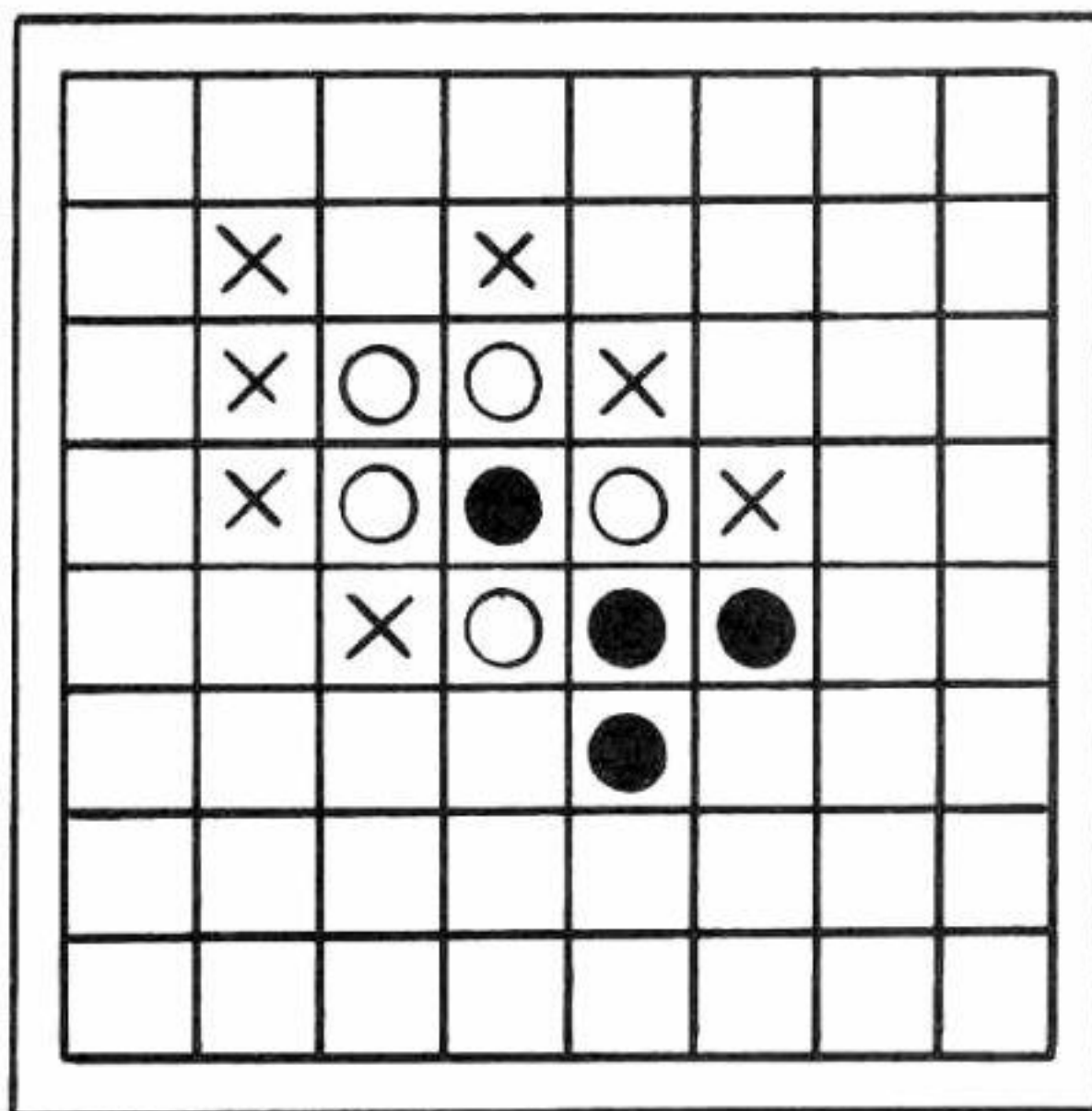
P3(): Ad ogni casella è assegnato un numero, dipendente dalla forza della posizione. Quando esiste possibiltà di scelta, il computer valuta quale ha il "peso" maggiore.

X,Y: Coordinate scacchiera. Per l'INPUT e il posizionamento della pedina.

GC,GC$: Mossa del giocatore.

PN: Numero pedine nere
PB: Numero pedine bianche

TU: Giocatore di turno
TU = 1 muove il nero
TU = 2 muove il bianco

A,I,Q,K: Variabili di comodo.

Tipica posizione di una fase iniziale di REVERSI. I giocatori depongono a turno una pedina, con lo scopo di catturare i pezzi nemici. Ad esempio, nella situazione in figura, il nero può muovere in una delle caselle contrassegnate con la crocetta. Tutti i pezzi bianchi che vengono a trovarsi tra due neri, vengono catturati, ma non rimossi dalla scacchiera: diventano a loro volta neri. Le catture possono essere fatte in senso orizzontale, verticale o diagonale. Oppure, se la mossa è abile, in più direzioni contemporaneamente.

Computer e intelligenza artificiale

Oggigiorno si sente parlare molto di macchine intelligenti, o addirittura pensanti. Il programma REVERSI, anche se non può essere paragonato ai più sofisticati sistemi esperti, è un valido esempio di "intelligenza artificiale".

Ben lontani dal giorno in cui si vedranno all'opera macchine veramente "pensanti" (inteso come pensiero umano), l'attuale tecnologia è invece in grado di costruire computer o programmi che abbiano almeno una competenza su un argomento ben delimitato. Programmi per gli scacchi, per la diagnosi delle malattie o per l'analisi geologica, possono a volte superare anche un abile professionista del settore.

Reversi non è certamente un campione, ma dà un'idea su come affrontare problemi anche ben più complessi:

a) studio dell'argomento: regole del gioco, mosse legali e non, casi di vittoria...

b) fase di analisi: valutazione della posizione, scelta della mossa più efficace...

Infine dobbiamo munire il programma delle necessarie interfacce per il colloquio con il giocatore: scacchiera, pedine, conteggio dei punti ecc.

Nel progettare il FLOW-CHART, può essere molto comodo seguire di pari passo le stesse tappe che noi, come persone, eseguiamo magari inconsciamente. Vediamo cosa succede se ad esempio decidiamo di fare una partita reale con un amico.

Innanzitutto bisogna prendere la scacchiera.

Qualche sorrisino ironico comparirà di fronte alla banalità di simile affermazione. Provate però a dire ad un computer: "Prendi gli scacchi!".

Una sezione di programma dovrà essere appositamente predisposta allo scopo: la routine DISEGNA SCACCHIERA compie questa funzione ed inoltre dispone i pezzi nella configurazione iniziale.

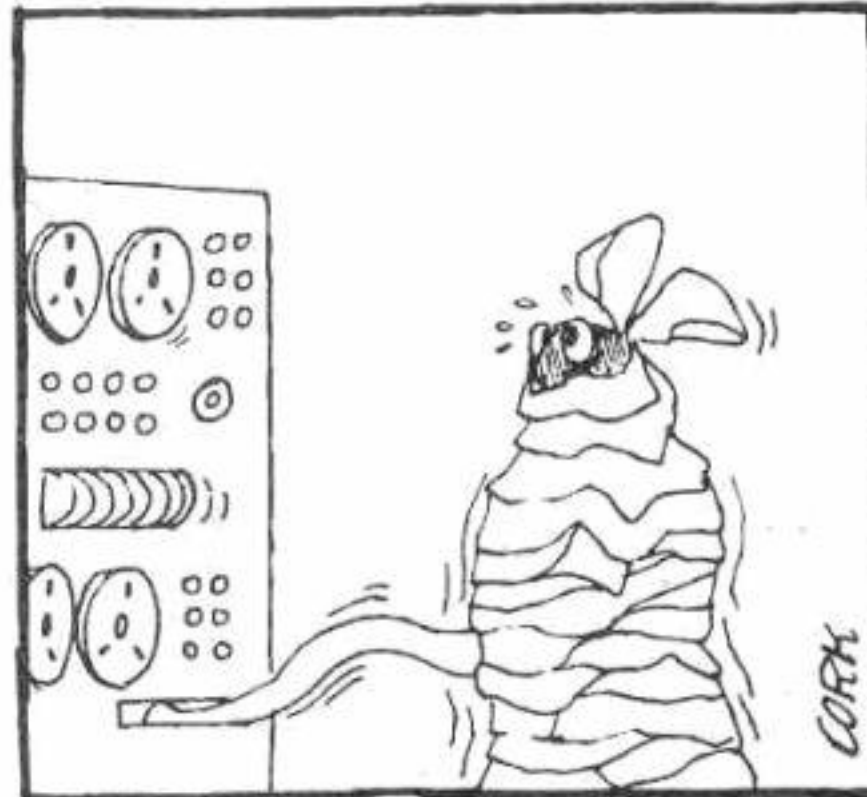
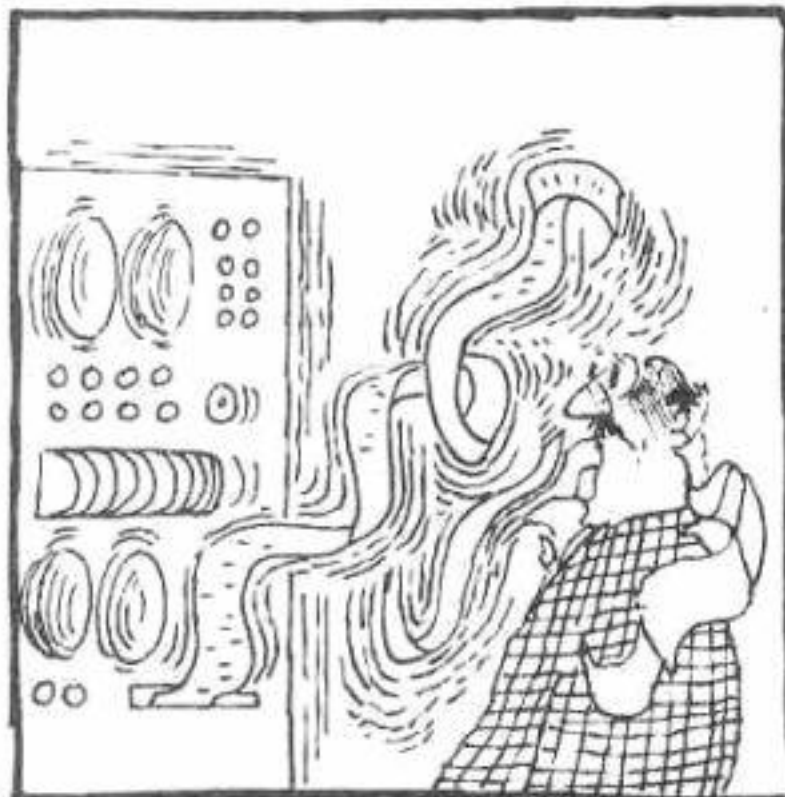
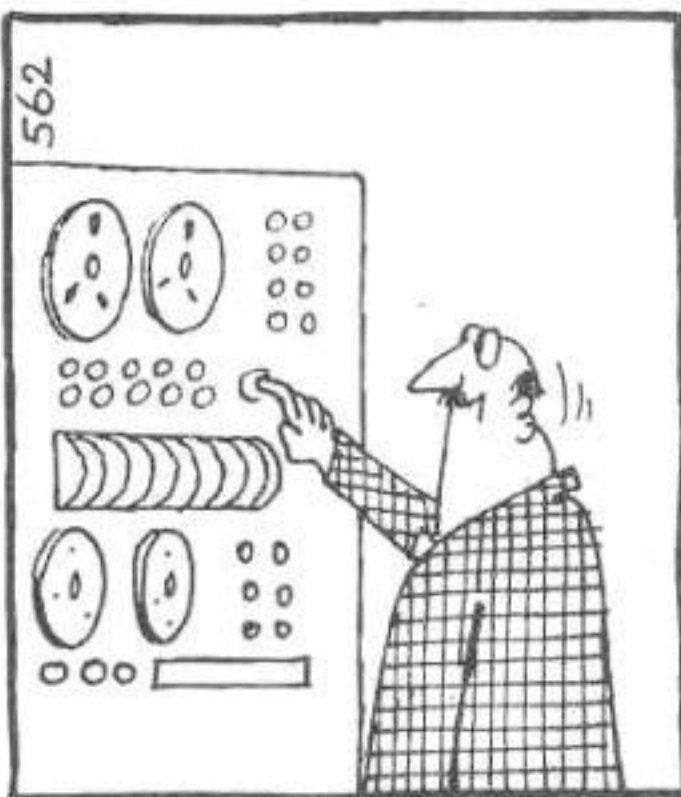
Chi gioca con i bianchi? Il computer sarà così gentile da lasciare a noi questa scelta (INIZIALIZZA).

Si può finalmente iniziare (INIZIO PARTITA). Tengo i bianchi e tocca a me muovere (INPUT). Studio un pò la situazione e decido dove mettere la pedina. Però il mio avversario sta bene attento a che io non compia mosse illegali o senza senso (servendosi di SUBROUTINES). Infine, se tutto è O.K., poso la pedina e, se è il caso, faccio le prese (manovra PEDINA).

Ora è il turno del computer: con rapide scorse alla scacchiera, stabilisce le possibili alternative e quale fra queste ha il "peso" maggiore (DATA SCACCHIERA) o è più conveniente (STRATEGIA).

```
100 REM  REVERSI
110 :
120 REM  C64 - C16 - PLUS4 - C128
130 :
140 REM  FLAVIO MOLINARI
150 :
160 REM  ************************
170 REM  *                      *
180 REM  *NOTA PER C16 E PLUS4: *
190 REM  *DIGITARE COLOR 0,7,5  *
200 REM  *E MODIFICARE LA LINEA *
210 REM  * 1920 IN:             *
220 REM  *                      *
230 REM  * 1920 RETURN          *
240 REM  *                      *
250 REM  ************************
260 :
270 REM  ************************
280 REM  *     INIZIALIZZA      *
290 REM  ************************
300 CLR
310 DIM  D$(24):FOR  Q=2  TO  24:D
    $(Q)=CHR$(19)+D$(Q-1)+CHR$(17)
    :NEXT
320 DIM  P1(80),P2(71),P3(71),P4(8
    0)
330 TU=2:CP=1:GOSUB 1850
340 PRINT CHR$(147)"N. GIOCATORI ?
     (1/2)"
350 GET  A$:IF  A$="2"  THEN 400
360 IF  A$<>"1"  THEN 350
370 UG=1:PRINT"SCEGLI LE PEDINE (B
    /N)"
380 GET  A$:IF  A$="N"  THEN  CP=2
    :GOTO 400
390 IF  A$<>"B"  THEN 380
400 GOSUB 980:REM  SCACCHIERA
410 FOR  Y=2  TO  5:FOR  X=2  TO
    5
420 READ  A:P4(Y*9+X)=A:NEXT:NEXT
430 REM  SITUAZIONE INIZIALE
440 P1(30)=2:P1(31)=1:P1(39)=1:P1(
    40)=2:PN=2:PB=2
450 REM  ************************
460 REM  *  INIZIO PARTITA      *
470 REM  ************************
480 SW=1
490 IF  TU=1  THEN  M$=CHR$(144)+"
    MOSSA: NERO  "
500 IF  TU=2  THEN  M$=CHR$(144)+"
    MOSSA: BIANCO"
510 PRINT D$(2) TAB(26);M$
520 IF  CP=1  THEN  AL=PN+1:GOTO 5
    40
530 AL=PB+1
540 PRINT D$(10) TAB(26)"NERI    "P
    N;CHR$(157)CHR$(32)
550 PRINT D$(15) TAB(26)"BIANCHI"P
    B;CHR$(157)CHR$(32)
560 IF  PB=0  OR  PN=0  OR  PB+PN=
    64  THEN 1440
570 IF  UG=1  AND  TU=CP  THEN  GO
    SUB 1530:GOTO 660
580 REM  ************************
590 REM  *       INPUT          *
600 REM  ************************
610 SP$="              ":REM  14 S
    PAZI
620 GC$="":PRINT D$(4) TAB(25) SP$
630 PRINT D$(4) TAB(30);:INPUT  GC
    $:GC=VAL(GC$)
640 IF  GC<1  OR  GC>64  THEN 610
650 Y=INT((GC-1)/8):X=GC-Y*8-1:XY=
    Y*9+X:IF  P1(XY)>0  THEN 610
660 IF  SW=0  THEN 770
670 REM  ************************
680 REM  *     METTE PEDINA     *
690 REM  ************************
700 IF  TU=1  THEN  PRINT CHR$(144
    );
710 IF  TU=2  THEN  PRINT CHR$(5);
720 PRINT D$(Y*3+2) TAB(X*3+1)PD$:
    GOSUB 1920
730 IF  P4(XY)=0  THEN 760
740 GOSUB 1220
750 IF  PD>0  THEN  GOSUB 1320:P1(
    XY)=TU:GOTO 850
760 PRINT D$(21) TAB(26)CHR$(31)"N
    ON VALIDA"
770 FOR  K=1  TO  8:GOSUB 1920:NEX
    T
780 FOR  K=1  TO  1000:NEXT
790 IF  SW=0  THEN 830
800 PRINT CHR$( 31);:REM  CANCELLA
     PEDINATO 710
810 PRINT D$(Y*3+2) TAB(X*3+1)PD$
820 PRINT D$(Y*3+2) TAB(X*3+1)CHR$
    (154)GC$
830 PRINT D$(21) TAB(25)SP$:PRINT
     TAB(25)SP$
840 GOTO 900
850 IF  TU=1  THEN  PN=PN+PD+1:PB=
```

```
     PB-PD:GOTO 870
 860 PB=PB+PD+1:PN=PN-PD
 870 FOR  Q=1  TO  8
 880 IF  XY+P5(Q)>-1  THEN  P4(XY+P
     5(Q))=1
 890 NEXT
 900 TU=3-TU
 910 GOSUB 940
 920 IF  XY=0  OR  XY=7  OR  XY= 63
      OR  XY=70  THEN  GOSUB 1700
 930 GOTO 480
 940 FOR  K=0  TO  71:P2(K)=0:NEXT:
     RETURN
 950 REM  ***********************
     **
 960 REM   *  DISEGNA SCACCHIERA
     *
 970 REM  ************************
     *
 980 C1$=CHR$(169):C2$=CHR$(127):C3
     $=CHR$(17)+CHR$(157)+CHR$(157)
 990 PD$=CHR$(18)+C1$+C2$+CHR$(146)
     +C3$+C2$+C1$
1000 A$=CHR$(176):C$=CHR$(171):D$=C
     HR$(173):E$=CHR$(96)
1010 FOR  Q=1  TO  7
1020 A$=A$+E$+E$+CHR$(178)
1030 C$=C$+E$+E$+CHR$(123)
1040 D$=D$+E$+E$+CHR$(177):NEXT
1050 FOR  Q=1  TO  9:B$=B$+CHR$(125
     )+CHR$(32)+CHR$(32):NEXT
1060 A$=A$+E$+E$+CHR$(174):C$=C$+E$
     +E$+CHR$(179):D$=D$+E$+E$+CHR$
     (189)
1070 PRINT CHR$(147)CHR$(144):PRINT
      A$
1080 FOR  Q=1  TO  7
1090 PRINT B$:PRINT B$:PRINT C$:NEX
     T
1100 PRINT B$:PRINT B$:PRINT D$;:PR
     INT CHR$(154)CHR$(19)
1110 FOR  K=0  TO  7:FOR  H=1  TO
     8
1120 G$=MID$(STR$(8*K+H),2)
1130 PRINT D$(3*K+2) TAB(3*H-2)G$:N
     EXT:NEXT
1140 PRINT D$(11) TAB(10)CHR$( 5 )P
     D$
1150 PRINT D$(11) TAB(13)CHR$(144)P
     D$
1160 PRINT D$(14) TAB(10)CHR$(144)P
     D$
1170 PRINT D$(14) TAB(13)CHR$( 5 )P
     D$
1180 RETURN
1190 REM  ************************
1200 REM  *     AGGIORNAMENTI     *
1210 REM  ************************
1220 PD=0:FOR  K=1  TO  8:L=1:V=0:X
     X=0
1230 V=V+P5(K):IF  XY+V>70  OR  XY+
     V<0  THEN 1270
1240 IF  P1(XY+V)=5  THEN 1270
1250 IF  P1(XY+V)=3-TU  THEN  XX=1:
     L=L+1:GOTO 1230
1260 IF  XX=1  AND  P1(XY+V)=TU  TH
     EN  GOSUB 1280
1270 NEXT:RETURN
1280 W=1:V=0
1290 V=V+P5(K):P2(XY+V)=TU
1300 W=W+1:IF  W<=L-1  THEN 1290
1310 PD=PD+W-1:RETURN
1320 FOR  I=0  TO  71
1330 IF  P2(I)=0  OR  P2(I)=5  THEN
      1400
1340 IF  TU=1  THEN  PRINT CHR$(144
     );
1350 IF  TU=2  THEN  PRINT CHR$(5);
1360 L=INT(I/9)
1370 PRINT D$(L*3+2) TAB((I-9*L)*3+
     1)PD$
1380 GOSUB 1920
1390 P1(I)=TU
1400 NEXT:RETURN
1410 REM  ************************
1420 REM  *     FINE PARTITA      *
1430 REM  ************************
1440 PRINT D$(20) TAB(26)"FINE PART
     ITA"
1450 FOR  K=1  TO  8:GOSUB 1920:NEX
     T
1460 PRINT D$(24) TAB(26)"ANCORA ?
     S/N"
1470 GET  A$:IF  A$="N"  THEN  END
1480 IF A$<>"S" THEN 1470
1490 GOTO 300
1500 REM  ************************
1510 REM  *      STRATEGIA        *
1520 REM  ************************
1530 M1=-32000
1540 FOR  XY=0  TO  71
1550 IF  P1(XY)>0  OR  P4(XY)=0  TH
     EN 1610
1560 GOSUB 1220:IF  PD=0  THEN 1610
```

```
1570 TT=PB+PN:QW=TT/8*PD+P3(XY)*(65
     -TT)/8
1580 IF  QW>M1  THEN  M1=QW:H1=XY:N
     EXT:GOTO 1620
1590 IF  M1=0  THEN  NEXT XY:GOTO 1
     610
1600 IF  QW/M1>.85  AND  QW/M1<1.15
      THEN  ZZ=INT(RND(1)*2):IF  Z
     Z=1  THEN  M1=QW:H1=XY
1610 NEXT
1620 IF  M1=-32000  THEN  SW=0:PD=0
1630 XY=H1
1640 GOSUB 940
1650 Y=INT(XY/9):X=XY-Y*9
1660 RETURN
1670 REM  ************************
1680 REM  *    SUBROUTINES       *
1690 REM  ************************
1700 IF  XY=7   THEN 1760
1710 IF  XY=63  THEN 1790
1720 IF  XY=70  THEN 1820
1730 FOR  I=9  TO  13:P3(I)=15-I:NE
     XT
1740 FOR  I=1  TO  37  STEP  9:P3(I
     )=6-INT(I/9):NEXT
1750 GOTO 1840
1760 FOR  I=6  TO  42  STEP  9:P3(I
     )=6-INT(I/9):NEXT
1770 FOR  I=16  TO 12  STEP -1:P3(I
     )=I-10:NEXT
1780 GOTO 1840
1790 FOR  I=54  TO  58:P3(I)=60-I:N
     EXT
1800 FOR  I=64  TO  28  STEP -9:P3(
     I)=INT(I/9)-1:NEXT
1810 GOTO 1840
1820 FOR  I=61  TO  58  STEP -1:P3(
     I)=I-55:NEXT
1830 FOR  I=69  TO  33  STEP -9:P3(
     I)=INT(I/9)-1:NEXT
1840 RETURN
1850 FOR  I=1  TO  8:READ  A:P5(I)=
     A:NEXT
1860 FOR  X=0  TO  71:READ  A:P3(X)
     =A:NEXT
1870 FOR  I=8  TO  71  STEP  9:P1(I
     )=5:NEXT
1880 RETURN
1890 REM  ************************
1900 REM  *       SUONO          *
1910 REM  ************************
1920 SI=54272:POKE  SI+24,15:POKE
     SI+6,247:POKE  SI+4,17
1930 POKE  SI+1,40:FOR  Q=1  TO  50
     :NEXT:POKE  SI+4,0
1940 RETURN
1950 REM  ************************
1960 REM  *   DATA SCACCHIERA    *
1970 REM  ************************
1980 DATA -10,-9,-8,-1,1,8,9,10
1990 DATA  16,-8,5,2,2,5,-8,16,0
2000 DATA  -8,-12,-2,-2,-2,-2,-12,-
     8,0
2010 DATA  5,-2,8,2,2,8,-2,5,0
2020 DATA  2,-2,2,1,1,2,-2,2,0
2030 DATA  2,-2,2,1,1,2,-2,2,0
2040 DATA  5,-2,8,2,2,8,-2,5,0
2050 DATA  -8,-12,-2,-2,-2,-2,-12,-
     8,0
2060 DATA  16,-8,5,2,2,5,-8,16,0
2070 DATA  1,1,1,1,1,0,0,1,1,0,0,1,
     1,1,1,1
```

Riservato agli insegnanti

Interrogazioni computerizzate

Come utilizzare il C64 per seguire meglio il ciclo di apprendimento di una classe

Il termine "interrogazione", in effetti, non è dei più moderni a causa della sua genericità. Da un po' di tempo si preferisce sostituirlo con "verifica", "esame del livello raggiunto", "valutazione del grado di apprendimento" e con altri termini che meglio illustrano le capacità dei singoli allievi.

La "Cooperativa A.P.E." di Gorizia (Aggiornamento Programmazione Educativa, Via Dante, 8 Tel. 0481/34169), tenendo conto dei limiti imposti da un computer, ha tuttavia sviluppato un programma in grado di aiutare considerevolmente il lavoro di un insegnante.

È infatti possibile affidare al calcolatore il compito di accertare se i contenuti di una lezione siano stati assimilati o se, al contrario, si riscontrano carenze di apprendimento. È inoltre possibile compiere statistiche sui componenti di un'intera classe allo scopo di decidere, o meno, l'approfondimento di un certo argomento forse non ben compreso dalla maggioranza degli allievi.

Il programma Progredisco

Il package applicativo provato ci ha favorevolmente impressionato grazie ai numerosi lati positivi che evidenzia:

- Istruzioni realmente chiare, riportate in modo semplice, idonee ad essere "digerite" anche dall'insegnante completamente... digiuno di informatica.
- Il libretto allegato, di quasi 90 pagine, inizia fornendo, addirittura, informazioni su come accendere il computer, sulla tastiera e, naturalmente, su come caricare il programma, non tralasciando nulla sulle operazioni da compiere. Non sono stati trascurati nemmeno i suggerimenti sul come prendere il disco ed inserirlo correttamente nel drive!
- La semplicità d'uso è reale dato che non solo è impossibile compiere errori ma, soprattutto, perché vi sono numerose schermate che aiutano docente e allievo ad utilizzare al meglio il programma.

La sequenza didattica

Il package consente di programmare intere lezioni (chiamate, più propriamente, "sequenze didattiche") ciascuna delle quali può occupare fino a 99 schermate.

L'insegnante, se non desidera ricorrere alle sequenze didattiche poste in commercio dalla stessa A.P.E. può programmare le proprie lezioni in modo piuttosto semplice.

In pratica si tratta di creare una sequenza di schermate (è possibile inserirvi perfino disegni ricorrendo ai caratteri semigrafici), che verranno in seguito gestite dal singolo allievo. Le videate possono essere informative o interattive. Queste ultime, ad esempio, propongono una domanda ed indicano un gruppo di risposte tra le quali una sola è quella valida.

A seconda della risposta fornita il programma può visualizzare una fase successiva (se la risposta è giusta) oppure proporre altre domande o argomentazioni che facciano riflettere nel caso sia stata fornita una risposta errata.

È quindi possibile strutturare un vero e proprio diagramma di flusso tenendo presente una casistica varia di risposte fornite esatte o errate.

La vera novità del package, a nostro avviso, consiste però nel fatto che, per ciascun allievo, è possibile memorizzare l'elenco completo delle domande rivolte e delle risposte fornite e, per l'intera classe, effettuare una statistica sugli argomenti meglio compresi o che risultano assimilati superficialmente. In base alla percentuale fornita l'insegnante può prendere decisioni diverse: se l'intera classe ha risposto in modo positivo commettendo pochissimi errori, può tranquillmente affrontare nuovi argomenti del programma che intende svolgere. Al contrario, se cioè il numero di allievi che hanno fornito risposte errate risulta elevato, potrà riprendere o approfondire argomenti trattati forse in modo superficiale.

Il miglior risultato, ovviamente, si può ottenere avendo a disposizione un laboratorio di informatica adeguatamente attrezzato con un congruo numero di Commodore 64 e disk drive (è prevista perfino l'uscita su stampante).

Il prezzo al pubblico

Il prezzo di listino del programma Progredisco è di duecentomila lire, decisamente contenuto se paragonato a package didattici per computer più grandi. Con quest'unico package, come già detto, l'insegnante è in grado di sviluppare per proprio conto sequenze didattiche di ogni tipo.

Sono comunque disponibili sequenze compilate da esperti che, dimenticavamo di dirlo, hanno maturato una notevole esperienza nel campo didattico. Alcune sequenze, inoltre, sono particolarmente adatte per portatori di lievi handicap.

Coloro che desiderano utilizzare sequenze già predisposte, troveranno un catalogo particolarmente fornito, economico e, ci hanno assicurato, in continuo ampliamento.

Alcuni esempi: compresione di lettura (L. 40.000). Concetti spazio temporali (L. 25.000). Uso dei sinonimi (L. 25000). Concetti di bisogni, beni, lavoro (L. 45.000). Il comune medioevale (L. 25.000). Educazione stradale (L. 25.000).

Ogni confezione contiene un manuale e uno o due dischi a seconda dei casi.

Ideali per valutare il grado di apprendimento dei bambini delle ultime classi delle elementari, molte sequenze possono essere utilizzate anche nelle prime classi della scuola media.

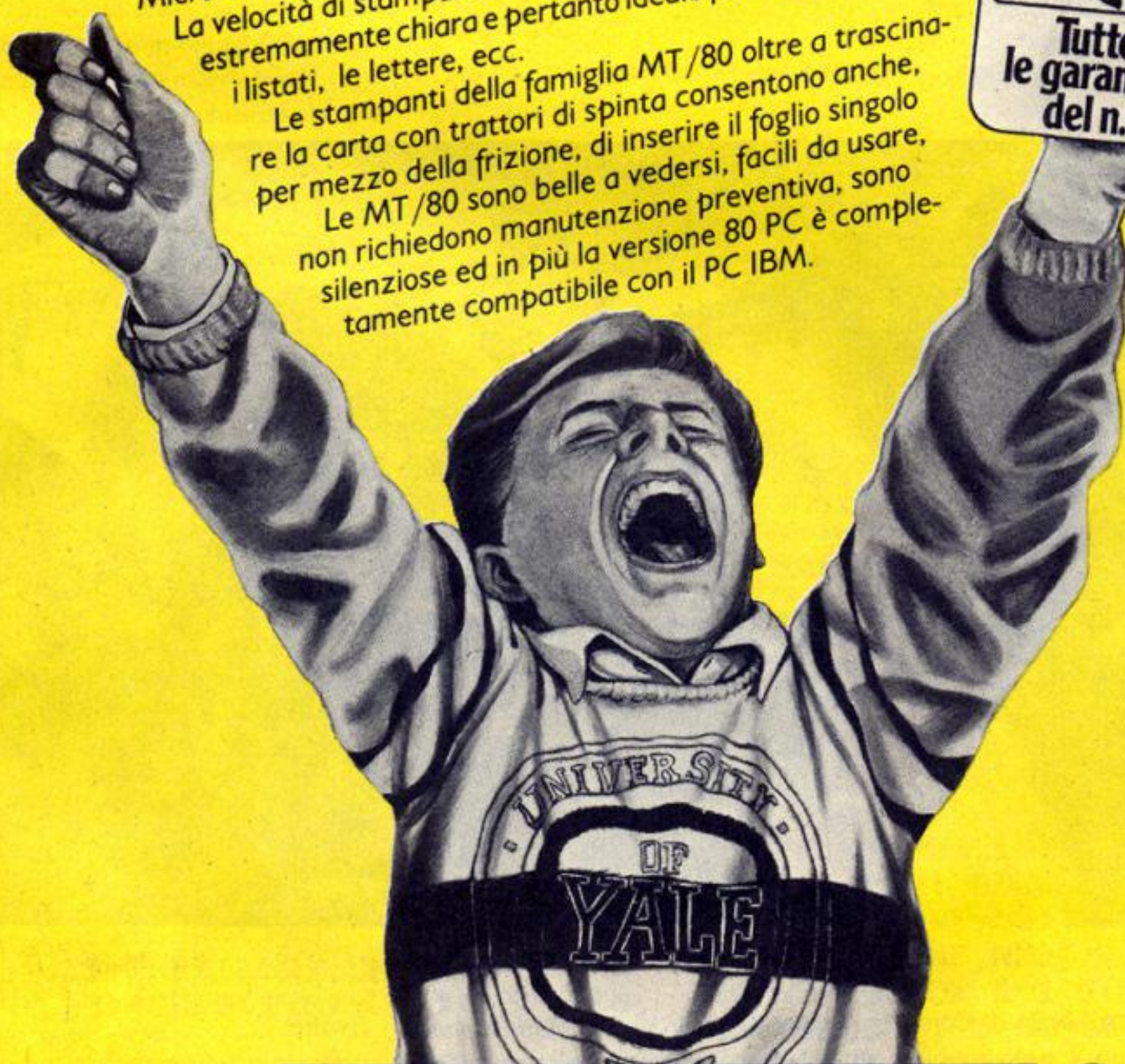
CON LA MT/80 SPENDENDO IL MINIMO HO IL MASSIMOOOO...!

Per **la stampa a basso costo**, le stampanti della famiglia MT/80 sono perfette, rispondendo ad ogni necessità di stampa. **La MT/80+ e la PC** dispongono di interfacciamento parallelo e seriale che permette di connettere questi prodotti a qualsiasi Micro o PC.

La velocità di stampa è di 100 o 130 cps. con una matrice estremamente chiara e pertanto ideale per stampare i listati, le lettere, ecc.

Le stampanti della famiglia MT/80 oltre a trascinare la carta con trattori di spinta consentono anche, per mezzo della frizione, di inserire il foglio singolo

Le MT/80 sono belle a vedersi, facili da usare, non richiedono manutenzione preventiva, sono silenziose ed in più la versione 80 PC è completamente compatibile con il PC IBM.

Tutte le garanzie del n. 1

MANNESMANN TALLY

20094 Corsico (MI) - Via Borsini, 6
Tel. (02) 4502850/855/860/865/870
Telex 311371 Tally I
00144 Roma - Via M. Peroglio, 15
Tel. (06) 5984723/5984406
10099 San Mauro (TO)
Via Casale, 308 - Tel. (011) 8225171
40050 Monteveglio (BO)
Via Einstein, 5 - Tel. (051) 832508

adverteam

QUALSIASI COMPUTER

Il gioco delle coppie

Verificate l'affinità esistente tra i componenti del gruppo di amici che frequentate più spesso.

di Di Giovanni Bellù

FRANCO ROSCIOLI

Qualche tempo fa, sul N.24 di Commodore Computer Club, è stato pubblicato un programma che simulava un test per le coppie. Usiamo il termine "simulare" perchè in realtà si trattava di una vera e propria beffa dato che il programma definiva come "ideale" la coppia scelta dal programmatore prima di sottoporre gli amici al test...

Visto quel simpatico programma, mi è subito venuto in mente un altro tipo di test, stavolta un po' più serio!

L'idea di fondo parte dal presupposto che due persone hanno una certa affinità se hanno gli stessi gusti. In considerazione di ciò, il programma di queste pagine formula un certo numero di domande le cui risposte sono fissate in numero di quattro.

Al test possono partecipare anche molte persone: penserà poi il computer ad individuare, tra tutti i partecipanti, coloro che hanno fornito risposte eguali al maggior numero delle domande rivolte.

Basta infatti associare, ad ogni risposta uguale, un punto di affinità. Se, pertanto, due persone rispondono in modo identico, ad esempio, a dieci domande su dodici, vuol dire che hanno dieci punti di affinità; mentre se forniscono la stessa risposta a due domande su dodici, ne totalizzano soltanto due.

Per calcolare qual'è il grado di affinità di due persone, sarà sufficiente individuare il numero di risposte uguali e rap-

portarle al numero delle domande poste.

In teoria l'affinità di coppia, come abbiamo detto, si incrementa con l'aumentare delle risposte uguali. In caso di risultati (per voi) disastrosi, potrete sempre affermare che vale esattamente l'opposto: totalizzare pochi punti di affinità, può voler dire avere caratteri opposti. Ora, dato che gli opposti si attraggono...

Non scoraggiatevi, quindi, se il computer riscontra una modesta affinità tra voi e la ragazza che preferite. Un po' di inganno (in buona fede) può risollevare unioni dichiarate fallimentari da un compromettente responso informatico!

Traete, invece, le conclusioni che più vi aggradano sulle elaborazioni che rappresentano una via di mezzo, come per esempio rispondere ugualmente a cinque oppure sei domande su un totale di 12.

Come gira il programma

Il programma richiede dapprima il numero di ragazze e di ragazzi che intendono sottoporsi al test, in seguito i loro nomi e infine, per ciascuno, porrà le domande previste indicando le quattro risposte possibili. Basterà premere i tasti da 1 a 4 per indicare la scelta e, naturalmente, il tasto Return.

Quando tutti avranno risposto alle domande, verranno visualizzati i nomi del ragazzo e della ragazza che hanno conseguito il maggior punteggio di affinità rispondendo in modo identico al maggior numero di domande.

Se si verificano punteggi eguali fra più coppie, verranno visualizzati i nomi di tutte queste.

In seguito, per ognuno verrà visualizzato il punteggio ottenuto con ciascuna persona del sesso opposto (con opzione su video oppure su stampante).

Come funziona il programma

Il programma è strutturato in modo tale da prevedere un qualsiasi numero di domande (compatibile ovviamente con la memoria del computer).

Nel listato di queste pagine, addirittura, ve ne sono appena due: per aggiungerne altre (insieme con le quattro risposte relative), basta che inseriate una nuova linea Basic prima di quella contenente i tre asterischi (che indicano la fine delle domande). Esempio:

XXXX DATA domanda, risposta1, risposta2, risposta3, risposta4

Ove XXXX indica il numero di linea e DATA l'istruzione Basic DATA.

Se una riga di programma non dovesse bastare, non c'è nessun problema, inserite una nuova linea iniziandola sempre con DATA.

Esempio:

1225 DATA PREFERISCI
MANGIARE?, CAVIALE,
ARAGOSTE,
POLENTA, PANINI

Il programma calcolerà automaticamente all'inizio il numero di domande inserite, purchè l'ultima istruzione DATA contenga sempre i tre asterischi.

```
100 REM   IL GIOCO
110 REM   DELLE COPPIE
120 :
130 REM   BY GIOVANNI BELLU'
140 REM   SEREGNO (MI)
150 :
160 ND=0:FOR I=1 TO 16:U$=U$+CH
    R$(32):NEXT:U1$=CHR$(32)+CH
    R$(146)
170 U2$=CHR$(18)+CHR$(150)+CHR$
    (32)+CHR$(159)
180 U3$=CHR$(150)+CHR$(18)+CHR$
    (32)+CHR$(146)+CHR$(157)+CH
    R$(159)
190 U4$=CHR$(157)+CHR$(172)+CHR
    $(157):Q$="PREMI UN TASTO P
    ER INIZIARE"
200 REM   CONTA DOMANDE
210 READ A$:IF A$="***" THEN 24
    0
220 FOR K=1 TO 4:READ A$:NEXT
230 ND=ND+1:GOTO 210
240 REM   ND=NUMERO DOMANDE INSE
    RITE
250 PRINTCHR$(147)"LE DOMANDE S
    ONO "ND
260 INPUT "NUMERO RAGAZZE";NF
270 IF NF<1 OR NF<>INT(NF) THEN
     250
280 INPUT "NUMERO RAGAZZI";NM
290 IF NM<1 OR NM<>INT(NM) THEN
     250
300 PRINTCHR$(147)"OK-, INSERIR
    E I NOMI DELLE RAGAZZE"
310 DIM F$(NF,ND),M$(NM,ND)
320 DIM PT(NF,NM)
330 NS=15:FOR Z=1 TO NF
340 GOSUB 1120:N$=LEFT$(N$+U$,1
    0):F$(Z,0)=N$
350 NEXT
360 PRINTCHR$(147)"OK-, INSERIR
    E I NOMI DEI RAGAZZI"
370 NS=15:FOR Z=1 TO NM
380 GOSUB 1120:N$=LEFT$(N$+U$,1
    0):M$(Z,0)=N$
390 NEXT
400 PRINTCHR$(147);"COMINCIANO
    LE RAGAZZE ..."
```

```
410 RT-5:GOSUB  1040
420 FOR Y-1 TO NF:RESTORE
430 PRINTCHR$(147)"TOCCA A "CHR
    $(18)F$(Y,0)U1$
440 PRINT:PRINT:PRINT:PRINTQ$
450 GOSUB 1070:FOR D-1 TO ND
460 PRINTCHR$(147);
470 READ A$:PRINTA$
480 FOR Z-1 TO 4:READ A$:PRINT:
    PRINT:PRINTZ;A$
490 NEXTZ:PRINT:PRINT
500 PRINT"RISPOSTA ";:NS-1:GOSU
    B 1120
510 IF N$<"1" OR N$>"4" THEN PR
    INT:GOTO 500
520 F$(Y,D)-N$
530 NEXTD,Y
540 PRINTCHR$(147);"ADESSO I RA
    GAZZI ..."
550 RT=5:GOSUB  1040
560 FOR Y=1 TO NM:RESTORE
570 PRINTCHR$(147)"TOCCA A "CHR
    $(18)M$(Y,0)U1$
580 PRINT:PRINT:PRINT:PRINTQ$
590 GOSUB 1070:FOR D-1 TO ND
600 PRINTCHR$(147);
610 READ A$:PRINTA$
620 FOR Z-1 TO 4:READ A$:PRINT:
    PRINT:PRINTZ;A$
630 NEXTZ:PRINT:PRINT
640 PRINT"RISPOSTA ";:NS-1:GOSU
    B 1120
650 IF N$<"1" OR N$>"4" THEN 50
    0
660 M$(Y,D)-N$
670 NEXTD,Y
680 REM  FINE INSERIMENTO RISPO
    STE
690 PRINTCHR$(147);"ADESSO ELAB
    ORO LE RISPOSTE:"
700 FOR Y-1 TO NF
710 FOR X-1 TO NM:PT-0
720 FOR D-1 TO ND
730 IF F$(Y,D)-M$(X,D) THEN PT-
    PT+1
740 NEXTD
750 PT(Y,X)-PT
760 NEXTX
770 NEXTY
780 REM  RICERCA COPPIA IDEALE
790 PT-0:FOR Y=1 TO NF:FOR X=1
    TO NM
800 IF PT(Y,X)>PT THEN PT-PT(Y,
    X)
810 NEXTX,Y
820 PRINTCHR$(147):PRINT"LA(E)
    COPPIA(E) IDEALE(I) E'(SONO
    ):"
830 FOR Y-1 TO NF:FOR X-1 TO NM
840 IF PT(Y,X)-PT THEN PRINTF$(
    Y,0)CHR$(211)M$(X,0)" PUNTI
    "PT
850 NEXTX,Y
860 PRINT:PRINT:PRINT"PREMI UN
    TASTO PER GLI ALTRI RISULTA
    TI":GOSUB 1070
870 PRINTCHR$(147);
880 PRINT:PRINT:PRINT"VIDEO O S
    TAMPANTE? (V/S)"
890 GET A$:IF A$-"V" THEN  VS-3
    :GOTO 920
900 IF A$="S" THEN  VS-4:GOTO 9
    20
910 GOTO 890
920 OPEN 1,VS:CMD 1
930 PRINTCHR$(18)" TEST VISTO D
    ALLE RAGAZZE:"
940 FOR Y-1 TO NF:PRINTCHR$(18)
    F$(Y,0)
950 FOR X=1 TO NM:PRINTM$(X,0)"
    = "PT(Y,X)
960 NEXTX:GOSUB 1070:NEXTY
970 REM  PRINTCHR$(147);
980 PRINT:PRINT:PRINTCHR$(18)"
    TEST VISTO DAI RAGAZZI :"
990 FOR X-1 TO NM:PRINTCHR$(18)
    M$(X,0)
1000 FOR Y-1 TO NF:PRINTF$(Y,0)"
     - "PT(Y,X)
1010 NEXTY:GOSUB 1070:NEXTX
1020 PRINT#1,:CLOSE 1
1030 RT-2:GOSUB 1040:PRINT"FINE
     TEST: AUGURI ....":END
1040 REM  RITARDO
1050 IF VS-4 THEN RETURN
1060 FOR R-1 TO 500*RT:NEXT:RETU
     RN
1070 REM  ATTENDE TASTO
1080 IF VS-4 THEN RETURN
1090 FOR I-0 TO 10:GET A$:NEXT:R
```

```
     EM  PULISCE BUFFER
1100 GET A$:IF A$="" THEN 1100
1110 RETURN
1120 REM
1130 N$=""
1140 FOR K=1 TO NS:PRINTCHR$(172
     );:NEXT
1150 FOR K=1 TO NS:PRINTCHR$(157
     );:NEXT
1160 GET A$:IF A$="" THEN 1160
1170 IF A$=CHR$(13) AND N$<>"" T
     HEN PRINTU2$:A=VAL(N$):RETU
     RN
1180 IF A$>="0" AND A$=<"9" THEN
      1230
1190 IF A$>="A" AND A$=<"Z" THEN
      1230
1200 IF A$="." OR A$="'" OR A$="
      " OR A$="-" OR A$="+" OR A
     $="(" OR A$=")" THEN 1230
1210 IF ASC(A$)=20 AND N$<>"" TH
     EN N$=LEFT$(N$,LEN(N$)-1):P
     RINTU4$;:GOTO 1160
1220 GOTO 1160
1230 IF LEN(N$)=NS THEN PRINTU3$
     ;:GOTO 1160
1240 N$=N$+A$:PRINTA$;
1250 IF LEN(N$)=NS THEN PRINTU3$
     ;:GOTO 1160
1260 GOTO 1160
1270 :    METTERE DA QUI IN POI L
     E
1280 :    DOM AND E E LE RISPOSTE
1290 :    (4 RISPOSTE PER OGNI DO
     M AND A
1300 :    E '***' ALLA FINE DEI D
     ATI.)
1310 DATA  CHE FILM PREFERISCI ?
     ,LOVE STORY,RAMBO,FANTOZZI,
     PROFONDO ROSSO
1320 DATA  CHE COLORE PREFERISCI
      ?,VERDE,BIANCO,ROSSO,GIALL
     O
1330 :
1340 :
60000 DATA  ***
63999 END
```

Il tuo schermo in Technicolor

Come assegnare un colore di sfondo diverso per ciascuna riga del video.

di Carlo & Lorenzo Barazzetta

Questo programma permette di utilizzare, ricorrendo sia a tecniche di Interrupt che all'uso del Background Color, alcune peculiarità della grafica del Commodore 64.

Molti sapranno di certo che digitando POKE 53281,0 il colore dello sfondo diventa nero.

Infatti il numero che segue la POKE è il cosiddetto indirizzo di memoria relativo al colore di sfondo, e il numero zero corrisponde al nero (vedi la tabella dei colori riportata nel manuale).
All'interno di un vostro programma Basic potete visualizzare, di norma, messaggi variamente colorati purchè, tutti, abbiano il medesimo colore di sfondo.

Col programma di queste pagine, invece, potrete usufruire, anzichè di uno solo, di ben 25 colori di sfondo, uno per ogni riga di testo.

Dopo aver digitato correttamente il programma N.1, salvatelo (su nastro, su disco oppure anche in turbo tape) e richiamatelo prima di caricare i vostri programmi che utilizzeranno la semplice routine in L.M. (Linguaggio Macchina).

Come si usa il programma

Dopo aver lanciato il listato N.1 con un semplice RUN, il computer risponderà col consueto READY se, ovviamente, non vi sono stati errori di copiatura.

A questo punto la routine in L.M. è stata trasferita in memoria e il programma in Basic non serve più.

Per abilitare questa routine battete SYS52026: noterete che il vostro video non avrà più un unico colore di sfondo, ma 25, uno per ogni riga di testo.

Mentre, prima, la locazione 53281 controllava il colore di sfondo, adesso le locazioni da 52000 a 52024 controllano, ognuna, il colore di sfondo di ogni singola riga del video. Se, ad esempio, digitiamo POKE 52000,0 il colore di sfondo della prima riga (quella più in alto) diventerà nero.

Il gioco applicativo

Per ulteriori chiarimenti potete digitare e provare il programma N.2 (dal titolo: Esempio sfondi) che fa apparire per un breve periodo di tempo un numero da indovinare.

La tecnica di "occultamento" del numero consiste, appunto, nell'aprire e chiudere, subito dopo, lo schermo con un effetto grafico decisamente inconsueto, paragonabile al movimento di una conchiglia.

Prima di utilizzare il gioco, come è intuitivo, è indispensabile caricare e lanciare il listato N.1.

Un ultimo avvertimento: se volete effettuare operazioni di I/O (input/output) con il nastro o con il disco dovrete disabilitare la routine proposta mediante un SYS 52123 e riassegnare il colore allo sfondo con POKE 53281, Colore. Ultimate le operazioni di I/O potrete riabilitare la routine sfondi con SYS 52026.

E' bene sapere, inoltre, che l'orologio del sistema (variabili TI$ e TI) non è più affidabile se la routine sfondi è abilitatà.

```
100 REM  SFONDI
110 REM  SCRITTO DA LORENZO BAR
    AZZETTA
120 REM  ABILITAZIONE SYS52026
130 REM  DISABILITAZIONE SYS521
    23
140 REM  SFONDI DA 52000 A 5202
    4
150 :
160 FOR A=0 TO 119:READ B
170 IF B>=0 AND B<256 THEN 190
180 PRINT"ERRORE NEI DATI";:END

190 C=C+B:POKE 52026+A,B:NEXT
200 IF C<>13824 THEN 180
210 :
220 DATA  120,169,127,141,13,22
    0,141,13
230 DATA  221,234,169,1,141,26,
    208,32
240 DATA  136,203,169,88,141,20
    ,3,169
250 DATA  203,141,21,3,88,96,21
    6,173,18
260 DATA  208,72,56,233,50,74,7
    4,74,170
270 DATA  189,32,203,141,33,208
    ,234,234
280 DATA  234,104,16,4,201,246,
    16,14,24
290 DATA  105,8,41,250,9,2,32,1
    38,203
```

```
300 DATA  234,76,126,234,32,136
    ,203,76
310 DATA  49,234,169,50,141,18,
    208,173
320 DATA  17,208,41,127,141,17,
    208,169
330 DATA  1,141,25,208,96,120,1
    69,49,141
340 DATA  20,3,169,234,141,21,3
    ,169,0
350 DATA  141,26,208,169,129,14
    1,13,220
360 DATA  88,96,-1
```

```
100 REM  ESEMPIO SFONDI
110 B=52000:REM  BASE PER I POK
    E
120 POKE B,7:REM  COLORE PRIMA
    RIGA
130 FOR A=1 TO 21
140 POKE B+A,0:REM  COLORE RIGH
    E 1-21
150 NEXT
160 FOR A=22 TO 24
170 POKE B+A,6:REM  COLORE RIGH
    E 22-24
180 NEXT
190 POKE 53280,8:REM  COLORE BO
    RDO
200 SYSB+26:REM  ABILITAZIONE S
    FONDI
210 PRINTCHR$(147);:REM  CANCEL
    LA VIDEO
220 POKE 646,9:REM  COLORE STAM
    PA
230 PRINT"RICORDATI IL NUMERO"
240 REM  AGGIUNGE UNA CIFRA
250 A$=A$+CHR$(RND(1)*9+49)
260 REM  POSIZIONAMENTO A META'
     VIDEO
270 PRINTCHR$(19);:REM  HOME
280 FOR A=1 TO 11:PRINTCHR$(17)
    ;:NEXT
290 POKE 646,0:REM  COLORE STAM
    PA
300 PRINTA$:REM  STAMPA IL NUME
    RO
310 REM  EFFETTO FINESTRA
320 FOR A=0 TO 10
330 POKE B+11-A,10:POKE B+11+A,
    10
340 NEXT
350 FOR A=10 TO 0 STEP -1
360 POKE B+11-A,0:POKE B+11+A,0
370 NEXT
380 REM  POSIZIONAMENTO A FINE
    VIDEO
390 FOR A=1 TO 11:PRINTCHR$(17)
    ;:NEXT
400 POKE 646,1:REM  COLORE STAM
    PA
410 INPUT "DIGITA IL NUMERO";B$
420 REM  TEST SUL NUMERO DIGITA
    TO
430 IF A$<>B$ THEN 480
440 REM  CANCELLAZIONE NUMERO D
    IGITATO
450 PRINTCHR$(145)SPC(18);
460 FOR A=1 TO 21:PRINTCHR$(32)
    ;:NEXT
470 B$="":GOTO 240
480 PRINTCHR$(19):REM  HOME
490 PRINT:PRINT
500 POKE 646,4:REM  COLORE STAM
    PA
510 PRINT"IL NUMERO ERA"CHR$(32
    )A$
520 PRINT"HAI TOTALIZZATO";
530 PRINTLEN(A$)-1"PUNTI"
540 POKE 646,2:REM  COLORE STAM
    PA
550 PRINT:PRINT"ANCORA ? (S/N)"
560 REM  INPUT RISPOSTA
570 GET A$:IF A$="S" THEN  RUN
580 IF A$<>"N" THEN 560
590 SYSB+123:REM  DISABILITAZIO
    NE SFONDI
600 POKE 53280,14:REM  COLORE B
    ORDO
610 POKE 53281,6:REM  COLORE SF
    ONDO
620 PRINTCHR$(147);:REM  CANCEL
    LA VIDEO
630 POKE 646,14:REM  COLORE STA
    MPA
640 PRINT"CIAO";
650 END
```

Enciclopedia di routine

a cura di Alessandro de Simone

13000 Scelta menu con cursore

(qualsiasi Commodore)

di Flavio Molinari

Realizzare un menu di scelta fra varie opzioni è piuttosto semplice. Un menu ben fatto, tuttavia, consente di valorizzare il programma in cui è sistemato tanto da conferirgli un aspetto "professionale" (anche in informatica l'estetica può avere la sua importanza).

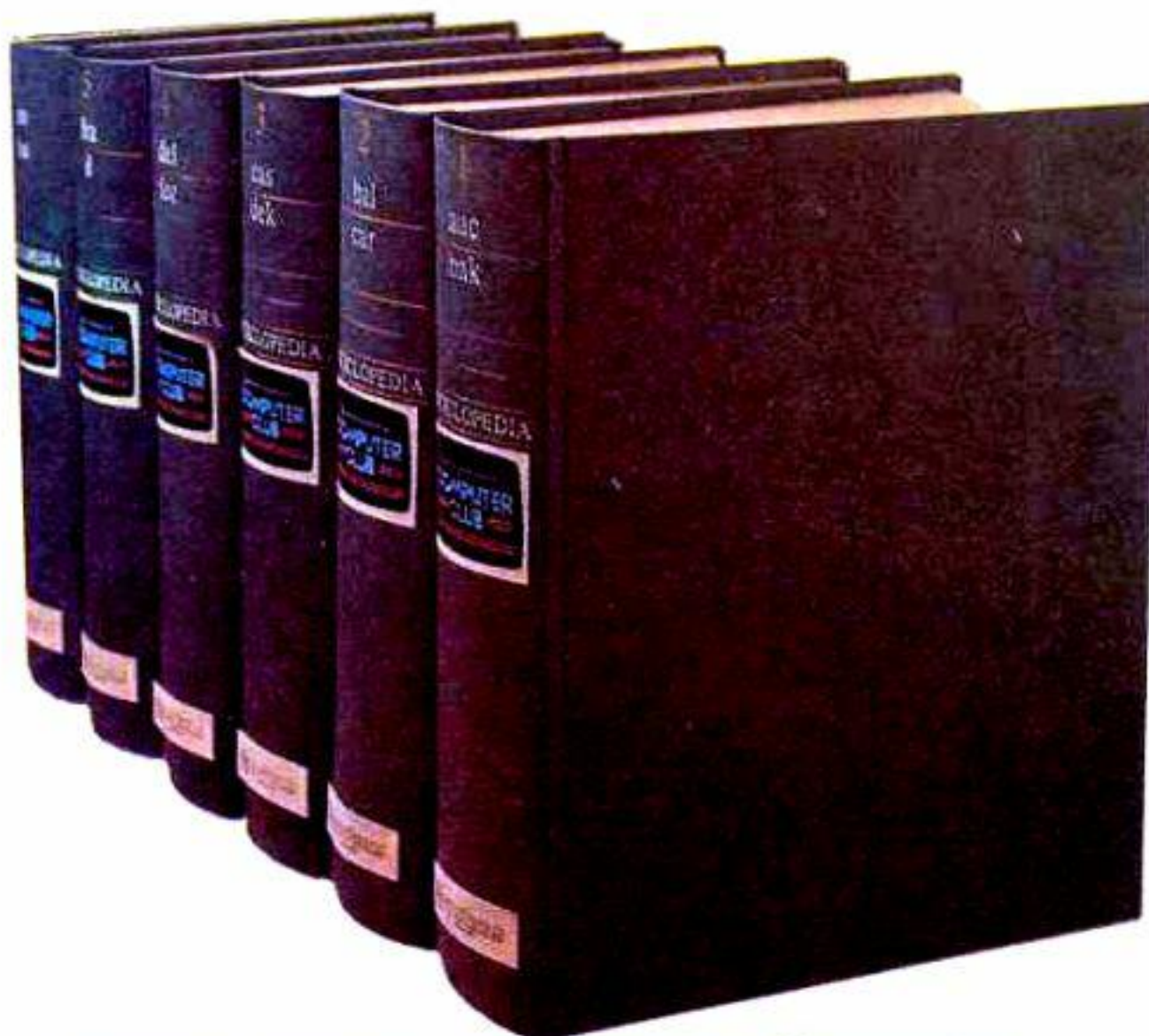
Il metodo più sbrigativo per realizzare un menu è senz'altro il seguente:

```
10 PRINT"1) OPZIONE 1"
20 PRINT"2) OPZIONE 2"
30 PRINT"3) OPZIONE 3"
40 INPUT"SCEGLI";SC$
```

ma, naturalmente, non vogliamo proporvene uno così banale.

Col programma pubblicato appare un menu composto da cinque opzioni la cui scelta si effettua spostandosi con il tasto cursore (altp/basso); contemporaneamente il messaggio selezionato compare in reverse. Questo modo di operare non solo renderà il menu esente da errori da parte dell'utilizzatore ma, soprattutto, consente di effettuare la scelta in modo più... vivace del solito INPUT o GET.

I messaggi di scelta vengono memorizzati nella matrice X0$(). L'uso di una variabile indicizzata, infatti, non crea problemi se l'indice rimane inferiore al valore 11. Non è infatti necessaria l'istruzione DIM che, al contrario, genererebbe un segnale di errore nel caso venga definita più di una volta.

Alla variabile X1 si assegna il numero di opzioni che si desidera visualizzare. Posizionato il cursore sulla "voce" scelta, si deve premere il tasto Return per tornare al programma principale. Da questo si può proseguire interrogando la variabile YV (che contiene il numero di opzioni scelto) con un'istruzione del tipo:

1000 ON YV GOTO 2000,3000,4000

Flavio Molinari

```
  100 REM  ESEMPIO D'USO
  110 REM  SCELTA MENU
  115 REM  MEDIANTE CURSORE
  120 REM  QUALSIASI COMMODORE
  130 :
  140 PRINTCHR$(147):REM  CLEAR S
      CREEN
  150 PRINT:PRINT:PRINT TAB(5)"GE
      STIONE CLIENTI"
  155 REM  OPZIONI
  160 X1$(1)="CARICAMENTO"
  170 X1$(2)="VARIAZIONE"
  180 X1$(3)="ANNULLAMENTO"
  190 X1$(4)="STAMPA"
  195 X1$(5)="FINE LAVORO"
  200 :
  210 X1=5:REM  NUMERO OPZIONI
  220 GOSUB 13000
  230 PRINTCHR$(19)"HAI SCELTO IL
       N."YV
  250 :
 9999 END
13000 Y1=10:REM  POSIZIONE ORIZZO
      NT.
13002 YV=1:XU$=CHR$(145):XD$=CHR$
      (17)
13005 PRINTCHR$(19):FOR XX=1 TO 6
      :PRINT:NEXT
13010 FOR XX=1 TO X1:PRINT TAB(Y1
      )X1$(XX):PRINT:NEXT
13015 PRINTCHR$(19):FOR XX=1 TO 6
      :PRINT:NEXT
```

```
13020 PRINT TAB(Y1)CHR$(18)X1$(YV
      )CHR$(146)
13025 GET XX$:IF XX$=CHR$(13) THE
      N RETURN
13030 IF XX$=XU$ AND YV>1 THEN GO
      SUB 13045:GOTO 13020
13035 IF XX$=XD$ AND YV<X1 THEN G
      OSUB 13050:GOTO 13020
13040 GOTO 13025
13045 YV=YV-1:PRINTXU$ TAB(Y1)X1$
      (YV+1):PRINTXU$XU$XU$;:RETU
      RN
13050 YV=YV+1:PRINTXU$ TAB(Y1)X1$
      (YV-1):PRINTXD$;:RETURN
13090 REM  X1$(): OPZIONI, MAX 10
        (X1)
13092 REM  YV: NUMERO SCELTA
13099 REM  SCELTA MENU' (CURSORE)
```

13100 Scelta menu con joystick

(C-64, C-16 e joystick)

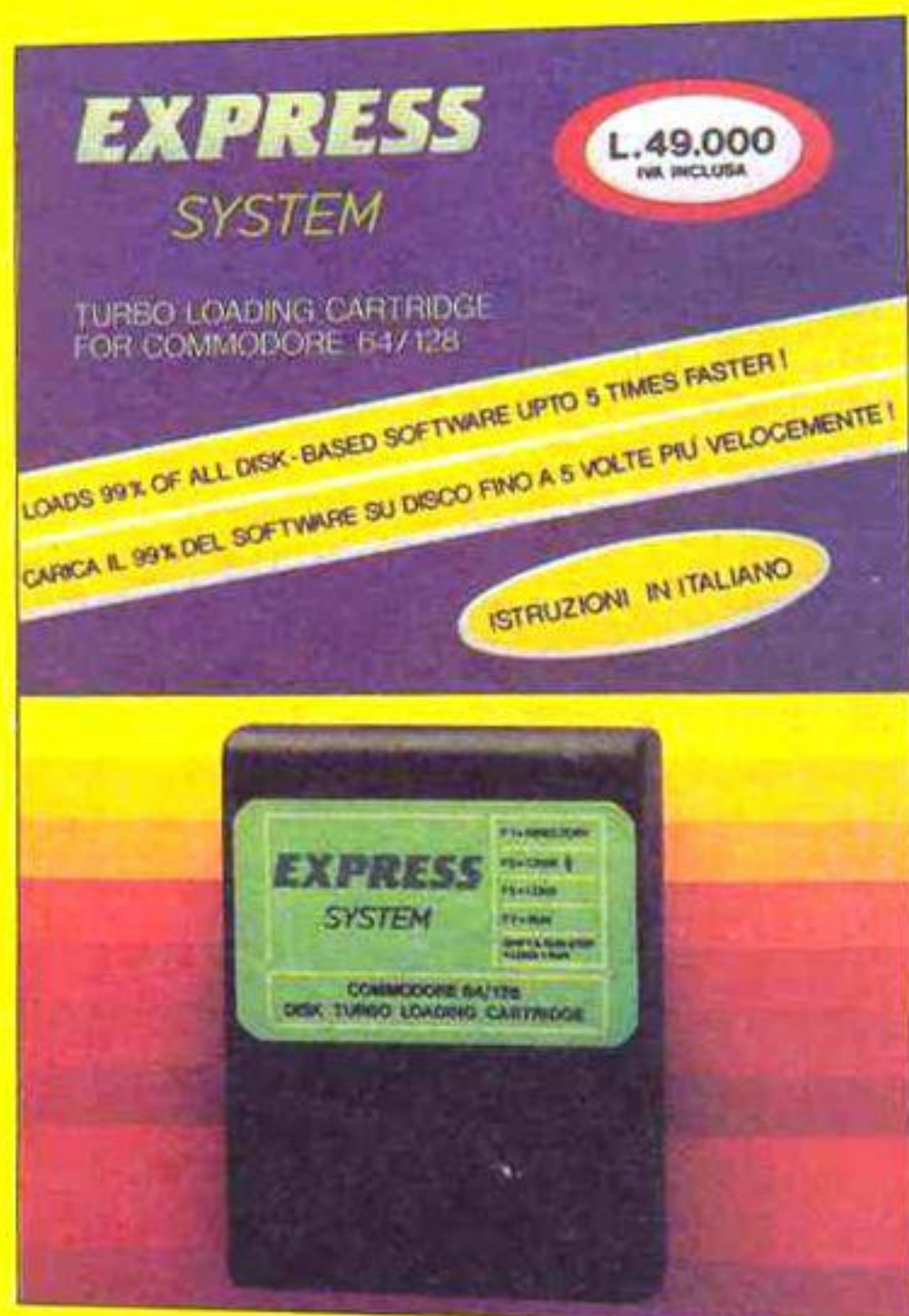
di Flavio Molinari

Analoga alla precedente, cambia unicamente l'aspetto operativo: la scelta si effettua muovendo il joystick anzichè premendo il tasto del cursore.

Le variabili interessate sono le stesse:
X0$(): voci del menu
X1: numero opzioni
YV: variabile che contiene, dopo la pressione del tasto Return, il numero di voce scelto.

Una volta posizionati sulla opzione desiderata, premere il pulsante Fire del joystick e proseguire nel programma in modo analogo a quello precedentemente descritto:
1000 ON YV GOTO 2000,3000 ...

Gli utenti del C-16 devono cancellare le istruzioni REM in modo da rendere operative le righe 13126, 13137 e 13138. Non devono invece trascrivere le righe 13125, 13130 e 13135, valide solo per il C-64.

Flavio Molinari

```
  100 REM  ESEMPIO D'USO
  110 REM  SCELTA MENU
  115 REM  CON JOYSTICK (PORTA 2)
  120 REM  COMMODORE 64, C-16
  130 :
  140 PRINTCHR$(147):REM  CLEAR S
      CREEN
  150 PRINT:PRINT:PRINT TAB(5)"GE
      STIONE DISCO"
  155 REM  OPZIONI
  160 X1$(1)="DIRECTORY"
  170 X1$(2)="COPIA FILE"
  180 X1$(3)="ERASE FILE"
  190 X1$(4)="RENAME"
  195 X1$(5)="FINE LAVORO"
  200 :
  210 X1=5:REM  NUMERO OPZIONI
  220 GOSUB 13100
  230 PRINTCHR$(147)"HAI SCELTO I
      L N."YU
  240 END
  250 :
13100 Y1=10:REM  POSIZIONE ORIZZO
      NT.
13102 YU=1:XU$=CHR$(145):XD$=CHR$
      (17)
13105 PRINTCHR$(19):FOR XX=1 TO 6
      :PRINT:NEXT
13110 FOR XX=1 TO X1:PRINT TAB(Y1
      )X1$(XX):PRINT:NEXT
13115 PRINTCHR$(19):FOR XX=1 TO 6
      :PRINT:NEXT
13120 PRINT TAB(Y1)CHR$(18)X1$(YU
      )CHR$(146)
13125 XJ=PEEK(56320):IF XJ=111 TH
      EN RETURN
13126 REM  XJ=JOY(2):IFXJ=128THEN
      RETURN
13130 IF XJ=126 AND YU>1 THEN GOS
      UB 13145:GOTO 13120
13135 IF XJ=125 AND YU<X1 THEN GO
      SUB 13150:GOTO 13120
13137 REM  IFXJ=1ANDYU>1THENGOSUB
      13145:GOTO13120
13138 REM  IFXJ=5ANDYU<X1THENGOSU
      B13150:GOTO13120
13140 GOTO 13125
13145 YU=YU-1:PRINTXU$ TAB(Y1)X1$
      (YU+1):PRINTXU$XU$XU$;:RETU
      RN
13150 YU=YU+1:PRINTXU$ TAB(Y1)X1$
      (YU-1):PRINTXD$;:RETURN
13190 REM  X1$(): OPZIONI, MAX 10
      (X1)
13195 REM  YU: NUMERO SCELTA
13199 REM  SCELTA MENU' (JOYSTYCK
      )
```

Elenco delle routine pubblicate

(Fra parentesi è riportato il numero di Commodore Computer lub su cui sono apparse)

63963 REM 12900 SCRITTA LAMPEGGIANTE (29)
63964 REM 12800 BORDO VIDEO IN TECHNICOLOR (29)
63965 REM 12700 FILL MEMORIA RAM (29)
63966 REM 12600 TEXT COPY (MPS 803) (29)
63967 REM 12500 CAMBIA COLORE PAG.TESTO (29)
63968 REM 12400 PRINT USING (29)
63969 REM 12300 M.C.D. e m.c.m. (29)
63970 REM 50500 VISUALIZZA FILE (28)
63971 REM 50400 LEGGE FILE RELATIVI (28)
63972 REM 50300 SCRIVE SU FILE RELATIVI (28)
63973 REM 50200 CREA FILE RELATIVI (28)
63974 REM 50000 LEGGE BLOCCHI LIBERI (28)
63975 REM 12200 NUMERI CONGRUI (28)
63976 REM 12100 PROTEZIONE SOFTWARE (28)
63977 REM 12000 KOALA (27)
63978 REM 11900 SCAMBIA PAGINA VIDEO (27)
63979 REM 11800 SALVA RAM (27)
63980 REM 11700 CALCOLATRICE (27)
63981 REM 11600 SCOMPOSIZ. SILLABE (27)
63982 REM 11500 CAR. HI—RES (27)
63983 REM 11400 ISTOGRAMMI (27)
63984 REM 50100 ESAME DIRECTORY (26)
63985 REM 11300 FUNZ.INV.IPERBOLICHE (26)
63986 REM 11200 FUNZ.INV. TRIGONOM. (26)
63987 REM 11100 FUNZIONI INVERSE (26)
63988 REM 11000 FUNZIONI IPERBOLICHE (26)
63989 REM 10900 CONVERSIONE DEC—ESA (26)
63990 REM 10800 CONTROLLO DATA (25)
63991 REM 10700 IMPULSI SONORI (25)
63992 REM 10600 REVERSE SCHERMO (25)
63993 REM 10500 INPUT CONTROLLATO (25)
63994 REM 10400 INCOLONNAME. VIRGOLA (25)
63995 REM 50000 N. BLOCKS FREE(DISCO) (24)
63996 REM 10300 INPUT & CONTR/DEFAULT (24)
63997 REM 10200 ESTRAZ.PAROLA DA FRASE (24)
63998 REM 10100 CAMBIA COL.BORDO/FONDO (24)
63999 REM 10000 CORNICE POLICROMA (24)

Simon's Basic

☐ **Perchè non parlate mai delle applicazioni del Simon' Basic per il C—64?**
(M. De Giorgi — Marcignago)

☐ **Il programma Simon's Basic si potrebbe inserire sulla memoria ROM del C—64 in modo da averlo disponibile subito all'accensione?**
(M. Brandstetter — Bolzano)

• Il programma Simon's Basic è un'utility che aggiunge nuovi potenti comandi Basic di semplice gestione per il Commodore 64. Fu inventato da un ragazzo, allora sedicenne, che in seguito ha creato una software house di nota fama.

Al giorno d'oggi, comunque, molti sono i programmi di utility prodotti, ed il Simon's deve ritenersi "quasi" obsoleto. Questo è il motivo per cui non parliamo spesso (anzi, mai) di programmi che lo utilizzano.

La prima versione del Simon's Basic era su cartuccia ma, in seguito, si preferì commercializzarlo su disco.

E' impossibile trasferire un programma, qualunque esso sia, su ROM perchè queste sono tipi di memoria a sola lettura (Read Only Memory). Si potrebbe tentare di riportare il programma, fatte le dovute modifiche, su memorie Eprom, ma la fatica che richiederebbe una simile operazione è di gran lunga superiore al costo del nuovo Commodore 128 che utilizza il Basic 7.0, molto più potente e versatile del Simon's Basic.

L'alternativa, ovviamente, è quella di rintracciare una delle prime versioni del Simons' su cartuccia.

E il C—128?

☐ **Molti lettori chiedono numerose informazioni sul nuovo Commodore 128 e, soprattutto, programmi che lo utilizzino in modo 128 e non 64.**

• Quando un nuovo computer viene posto in vendita è necessario un periodo di "rodaggio" per esaminare il successo conseguito e le specifiche richieste dei suoi nuovi utenti.

I lettori di C.C.C. stiano pure tranquilli: Basta avere pazienza e anche per questo computer sarà pubblicato software specifico.

Modifiche da apportare

Molti lettori, soprattutto nuovi, chiedono le modifiche da effettuare per far girare, sui propri calcolatori, programmi specifici di altri computer.

• Abbiamo più volte ribadito che è molto, molto difficile modificare programmi, specie se in Linguaggio Macchina (L.M.) Tali difficoltà sono principalmente dovute ai seguenti motivi:

• Gli effetti sonori dei tre computer (C—64, Vic—20, C—16) sono dovuti, rispettivamente, a circuiti integrati profondamente diversi l'uno dall'altro. Il modo di programmarli, quindi, varia di conseguenza.

• Gli sprite, protagonisti principali di numerosi videogiochi del C—64, non possono esser riprodotti in alcun modo sul C—16 oppure sul Vic—20.

• La gestione interna della memoria è diversa nei tre computer e questo fatto rende praticamente impossibili le modifiche da effettuare.

• Il video del Vic—20 è di formato 22 righe per 23 colonne, mentre quelli del C—16 e C—64 sono di 25 righe per 40 colonne. Nel visualizzare alcune schermate, di conseguenza, possono capitare errati allineamenti. Se ciò non bastasse, lo scrolling del video è gestito diversamente nel C—64 e nel C—16.

• Solo programmi in "puro" Basic, che cioè non richiedono visualizzazioni particolari (e che non contengono istruzioni del tipo POKE, PEEK, SYS) possono esser fatti girare allo stesso modo (o quasi) nei diversi tipi di computer.

• Aumentando la memoria RAM del Vic 20 questo non si trasforma in C—16. Allo stesso modo il C—16, pur se espanso al massimo, non "diventa" nè un Plus/4 nè un C—64.

Teo Rusconi ha appena sfatato la leggenda secondo la quale i floppy disc sono tutti uguali

Difatti sembrano tutti uguali finchè non si osserva con attenzione il jacket. Qui termina l'uguaglianza.

La maggior parte delle società costruttrici sigillano i dischi un punto qui, un punto là, lasciando parte dei lembi non sigillati.

Prima o poi ai lembi accadono cose naturalissime: si gonfiano, si curvano, si raggrinziscono... in poche parole si aprono.

Con penne, matite, unghie persino un ragazzino di quattro anni come Teo può infilarsi in quegli spazi aperti.

Naturalmente è un danno enorme perchè se si inserisce qualcosa di molle e slabbrato nel disc-drive quest'ultimo può incepparsi; si può rovinare la testina e si possono perdere i dati. Questo può accadere con gli abituali sistemi di chiusura ma non con i dischetti Memorex che usa un procedimento esclusivo chiamato "Solid-Seam Bonding".

Con questo sistema ogni singolo millimetro quadrato dei lembi di tutti i dischi Memorex viene sigillato ermeticamente, rendendoli più rigidi e più resistenti.

È un sistema che consente al floppy disc di sostenere ogni assalto, che impedisce alla testina di rovinarsi e ai dati di andare perduti.

Il che sta a dimostrare che un floppy disc Memorex non è uguale a tutti gli altri: è migliore. E il sistema di saldatura è solo un esempio della cura infinita con cui viene prodotto ogni floppy disc Memorex; sia esso da 8", da 5 1/4" o il nuovo 3 1/2". Questa estrema accuratezza dà la garanzia che ogni disco Memorex è al 100% perfetto.

La prossima volta che acquistate un floppy disc - o qualche centinaio - ricordate: non tutti i dischetti sono uguali...
Memorex vi mette al riparo da qualsiasi inconveniente.

MEMOREX MEMOREX MEMOREX

è importante scegli
MEMOREX
A Burroughs Company

BURROUGHS-MEMOREX S.P.A.
Divisione Computer Media
Via Ciro Menotti, 14 Tel. 02/718551
20129 MILANO MI

SCAMBIATEVI LE LISTE

Gianluca Riccardi/Costantino Mastracco - Via P.D. Passionista 3B - 03024 Ceprano (Fr) Tel. 0775/950102

Bruno Castelnuovo - Via S. Antonio 3 - 23020 Montagna in Valtellina (So) - Tel. 0342/ 380234

Daniele Peruzzi - Via J. Cook 8 - 52100 Arezzo Tel. 0575/911306

Eugenio Ferla - Via P. Cominio 56 - 00175 Roma - Tel. 06/765535

Bernadotti Flavio - Via Trento 10 - 15040 Montecastello (Al)

Fabio Bellassai - Via F.lli Bandiera 13 - 95100 Catania - Tel. 415353

Sebastiano Caramagno - Contrada Cipollazzo 96011 Augusta - Tel. 0931/993333

Erminio Dessì - V.le del Lavoro 20 - 87012 Castrovillari (Cs) - Tel. 0981/46393

Riccardo Donadon - Via Postumia 22 - 31055 Quinto di Treviso - Tel. 0422/20894

Antonio Conticello - Via Galli della Mantica 28 - 00122 Ostia (Rm) - Tel. 06/5603759

Michele Cervelli - Via Fiorentina 43 - 56025 Pontedera (Pi) - Tel. 0587/56046

Enzo Landro - Via delle Terme 97 - 95024 Acireale (Ct) - Tel. 095/608294

Riccardo Badenghini - Via Pianezza 153 - 10151 Torino - Te. 011/734940

Paolo Pagani - Via Guillenzani 5 - 41100 Modena - Tel. 059/350964

Simone Merlini - Via Baglioni 24 - 06100 Perugia - Tel. 075/61580

Oscar Bresciani - Via Veneto 24 - 20031 Cesano Maderno (Mi)

Gerardo Marra - Via P. Baratta 91/A - 84091 Battipaglia (Sa) - Tel. 0828/23130

Francesco Saracino - Via E. Saracino 25 - 70032 Bitonto (Ba) - Tel. 617600

Marco Cavedoni - Via G. Pascoli 95 - 41058 Vignola (Mo) - Tel. 059/771734

Salvatore Spanedda - C.so Umberto - 09073 Gugliera - Tel. 0785/39857

Sandro Garzelli - Via G. Amendola 17 - 50053 Empoli (Fi) - Tel. 0571/78361

Fràbrizio Rizzi - Via Castello 3060B - 30122 Venezia - Tel. 041/22883

Marco Mori - Via F. Crispi 13 - 52100 Arezzo - Tel. 0575/22013

Primo Valmori - Via della Resistenza 18 - 48020 Ca' di Lugo (Ra)

Roberto Pallotti - Via Audinot 35 - 40134 Bologna - Tel. 051/421956

Umberto Morra di Cella - Via Gen. Perotti 23 10091 Alpignano (To) - Tel. 011/9677679

Massimiliano Gucciardi - Via Brecce Bianche 80 - 60131 Ancona - Tel. 071/862641

Antonio Santagata - Via Valbruna 16 - 33100 Udine - Tel. 43456

Igor Princic - V.le XX Settembre 35 - 34137 Gorizia - Tel. 0481/89200

Massimo Fortin - Via Rovetta 6 - 20127 Milano - Tel. 2852865

Poalo Solaro - P.zza Medaglie d'Oro 13 - 14100 Asti - Tel. 0141/51973

Roberto Michelotti - Via G.B. Gaulli 11/23 - 16143 Genova

Maurizio Belloni - Via Legionari Polonia 33 - 24100 Bergamo - Tel. 035/262083

Elisa Leoni - P.zza Giotto 20 - 52100 Arezzo - Tel. 0575/20606

Edoardo Lampis - Via Cadorna 21 - 50129 Firenze - Tel. 055/494648

Stefano Sampaolesi - Via A. Moro 15 - 63021 Amamdola (Ap) - Tel. 0736/97462

Albano Filiaci - Via B. Miriam 61 - 63035 Offida (Ap) - Tel. 0736/80144

Marco Milesi - Via Dante 55 - 24032 Calolziocorte (Bg) - Tel. 0341/645507

Massimiliano Mattioli - Via G. Paganelli 10 - 47100 Forlì - Tel. 0543/51261

Michele Melchionda - Via Tagliamento II Traversa 6 - 83100 Avellino - Tel. 0825/38091

Roberto Oberto - Via Biscaretti 7 - 10025 Pino Totinese (To) - Tel. 011/842357

Roberto Guida - P.zza Ernesto Cesaro 66 - 80058 Torre Annunziata - Tel. 081/8616561

Angelo Germinario - Via F. Fusco 36 - 95100 Catania - Tel. 095/437234

Andrea Franci - Via A. de Gasperi 34 - 52100 Arezzo - Tel. 0575/33540

Giorgio Franchi - C.so Quintino Sella 87 - 10132 Torino - Tel. 011/832875

Marco Maggi - Via Serlio 8/2 - 20139 Milano

Roberto Boselli/Andrea Padovani - Via Gonzales 4 - 20139 Milano - Tel. 02/567465/ 5692965

Sergio Rivoletti - Via Tevere - 63017 Porto San Girogio - Tel. 0734/4419

Marco Arcari - Via Walter Muttini 14 - 54036 Massa Carrara (Ms)

Claudio Luciani - Via J. Gagarin 34 - 63020 Piane di Falerone

INVIARE TUTTA LA PAGINA ANCHE SE SI UTILIZZA UNA SOLA SCHEDA

Nome .... Cognome ....

Via .... N° .... CAP. .... Città ....

Telefono .... Orario ....

Registrate il mio abbonamento annuale a Commodore Computer Club.

☐ Ho versato oggi stesso il canone di L. 35.000 a mezzo c/c postale n° 37952207 intestato a:
Systems Editoriale - V.le Famagosta, 75 - 20142 Milano

☐ Ho inviato oggi stesso assegno bancario n............
per l'importo di L. 35.000 intestato a Systems Editoriale
Si prega di scrivere il proprio nome e l'indirizzo completo in modo chiaro e leggibile. Inviare la fotocopia del bollettino di c/c postale.

Considerando che i numeri 1, 2 e 7 sono esauriti, vogliate inviarmi i numeri arretrati .... al prezzo di L. 5.000 cadauno per richieste fino a 4 numeri, o di L. 4.000 cadauno per richieste oltre i 4 numeri arretrati, e perciò per un totale di L.............. Sono a conoscenza che i fascioli suddetti non saranno inviati in contrassegno e, pertanto, ho provveduto oggi stesso a versare il canone di L.................. a mezzo c/c postale n. 37952207 intestato a:
Systems Editoriale - V.le Famagosta, 75 - 20142 Milano

STATISTICA

Non posseggo un computer ☐

Posseggo un C64	si ☐	... no ☐
Posseggo un VIC 20	si ☐	... no ☐
Posseggo un Commodore Plus 14	si ☐	... no ☐
Posseggo un Commodore Plus 16	si ☐	... no ☐
Posseggo un registratore dedicato	si ☐	... no ☐
Posseggo un drive 1541	si ☐	... no ☐
Posseggo una stampante	si ☐	... no ☐
Posseggo un monitor	si ☐	... no ☐

COLLABORAZIONE

A titolo di prova vi invio un articolo e la cassetta........................disco....................................
col programma che intendo proporre per la pubblicazione di cui garantisco l'originalità.

DOMANDA/RISPOSTA ....

....

....

....

....

....

RICHIESTA ARGOMENTI

Mi farebbe piacere che Commodore Computer Club parlasse più spesso dei seguenti argomenti:

1/ ....

2/ ....

3/ ....

4/ ....

GIUDIZIO SUI PROGRAMMI DI QUESTO NUMERO

Ho assegnato un voto da 0 a 10 ai programmi che indico di seguito:

A/ .... Voto ....

B/ .... Voto ....

C/ .... Voto ....

D/ .... Voto ....

PICCOLI ANNUNCI ....

....

....

....

....

....

CERCO/OFFRO CONSULENZA ....

....

....

....

....

....

INVIARE IN BUSTA CHIUSA E AFFRANCANDO SECONDO LE TARIFFE VIGENTI A:

COMMODORE COMPUTER CLUB

V.le Famagosta, 75
20142 Milano

INVIARE TUTTA LA PAGINA ANCHE SE SI UTILIZZA UNA SOLA SCHEDA

Nome ____________ Cognome ____________

Via ____________ N° ______ CAP. ______ Città ____________

Telefono ____________ Orario ____________

24 ORE SU 24 DI MUSICA IN STEREOFONIA CON

CONCESSIONARIA
PER LA PUBBLICITÀ DI MILANO

RADIANT S.P.A.

CONCESSIONARIA
PER LA PUBBLICITÀ DEL CIRCUITO

gamma color italia S.r.l.

PALAZZO CANOVA CENTRO DIREZIONALE MILANO 2 - 20090 SEGRATE (MI)
TEL. 02/2155714 - 2155726 - 2155734

LOMBARDIA

Milano	95.9-92.8-97.1
Bergamo	99.3
Brescia	92-92.7
Como	97.1
Cremona	99.3
Pavia	95.9-97.1
Varese	94.9

LIGURIA

Genova	96.25
La Spezia	98.7

EMILIA ROMAGNA

Bologna	88.7
Piacenza	97.1

PIEMONTE/VAL D'AOSTA

Alessandria	104.3
Cuneo	90.6-97.6
Novara	97.1
Aosta	91.8-92

TOSCANA

Firenze	97.6-104.4
Livorno	98.2-97.3-100.6
Massa C.	98.7
Pistoia	97.6-104.4
Pisa	97.3
Lucca	97.3

LAZIO

Roma	99.5

systems